Preface: SCiL 2020 Editors' Note

Allyson Ettinger[1], Gaja Jarosz[2], and Max Nelson[2]
[1]University of Chicago and [2]University of Massachusetts Amherst

This volume contains research presented at the third annual meeting of the Society for Computation in Linguistics (SCiL), held in New Orleans, January 2-5, 2020, in conjunction with the annual meeting of the Linguistic Society of America.

Research was submitted to be reviewed either in the form of a paper, or as an abstract. The oral presentations, or talks, at the conference included both papers and abstracts. The papers presented as talks are the first 17 papers listed in the proceedings, and the remaining 23 papers were presented as posters. Authors of accepted abstracts were given the option of publishing an extended version; these follow the papers, and are themselves followed in this volume by abstracts that are published in their original two-page length.

In total, we received 82 submissions to the conference, 30 abstracts and 52 papers. 21 submissions were selected for oral presentation (26%) and 35 for poster presentation (43%). We asked authors to provide gender information on a voluntary basis at submission time. Of the submissions for which we received gender information (72), 39 (54%) included a female author. 27 (48%) of the accepted submissions included a female author, and 10 (56%) of the submissions accepted as talks included a female author.

We thank our reviewers for their indispensable help in selecting the research for presentation at the conference:

> Adam Albright, Eric Bakovic, Timothy Baldwin, Michael Becker, Emily M. Bender, Leon Bergen, Oliver Bonami, Sam Bowman, Jonathan Brennan, Lucas Champollion, Jane Chandlee, Rui Chaves, Alexander Clark, Jennifer Culbertson, Robert Daland, Philippe de Groote, Brian Dillon, Ewan Dunbar, Daniel Edmiston, Micha Elsner, Robert Frank, Richard Futrell, Matt Goldrick, Kyle Gorman, Thomas Graf, John Hale, Yiding Hao, Bruce Hayes, Jeff Heinz, Nick Huang, Tim Hunter, William Idsardi, Adam Jardine, Roni Katzir, Racy King, Christo Kirov, Andras Kornai, Sandra Kuebler, Andrew Lamont, Tal Linzen, Giorgio Magri, Fred Mailhot, Rob Malouf, Andrea E. Martin, Kevin Mcmullin, Emily Morgan, Aleksei Nazarov, Max Nelson, Tim O'Donnell, Alexis Palmer, Martha Palmer, Ellie Pavlick, Lisa Pearl, Laurel Perkins, Christopher Potts, Omer Preminger, Brandon Prickett, Ezer Rasin, Siva Reddy, James Rogers, Asad Sayeed, Thomas Schatz, Nathan Schneider, Andrea Sims, Caitlin Smith, Edward Stabler, Mark Steedman, Tom Wasow, Bonnie Webber, Aaron Steven White, Adina Williams, Colin Wilson

Thanks also to Tom Maxfield and Joe Pater for logistical help.

SCiL 2020 also included a plenary session on "Computation and Meaning" with invited talks by Ellie Pavlick (Brown University) and Christopher Potts (Stanford University). As part of SCiL 2020 there was also an NSF-funded workshop on "Formal Language Theory in Linguistics", which included several special sessions: a keynote address by Jeffrey Heinz (Stony Brook University), a special abstract-reviewed 'works in progress' session of talks, tutorials by Alëna Aksënova (Stony Brook University) and Kyle Gorman (CUNY), and mentoring events aimed at young researchers in computational linguistics. Further information can be found at our website: https://blogs.umass.edu/scil/scil-2020/.

The stability of segmental properties across genre and corpus types in low-resource languages

Uriel Cohen Priva[*] and **Shiying Yang**[*] and **Emily Strand**
Brown University
Department of Cognitive, Linguistic, and Psychological Sciences
190 Thayer St., Providence, RI 02912, USA
uriel_cohen_priva,shiying_yang,emily_strand@brown.edu

Abstract

Are written corpora useful for phonological research? Word frequency lists for low-resource languages have become ubiquitous in recent years (Scannell, 2007). For many languages there is direct correspondence between their written forms and their alphabets, but it is not clear whether written corpora can adequately represent language use. We use 15 low-resource languages and compare several information-theoretic properties across three corpus types. We show that despite differences in origin and genre, estimates in one corpus are highly correlated with estimates in other corpora.

1 Introduction

One of the challenges facing corpus research in phonology is the absence of detailed cross-linguistic phonological corpora. When a phonological trend is found in a language or a language family, e.g. OCP in Semitic (McCarthy, 1986), does it extend to other languages too? Variation-friendly versions of Optimality Theory (e.g. Anttila, 1997; Boersma, 1998; Goldwater and Johnson, 2003) predict that obligatory constraints in one language would appear as trends in other languages too, e.g. languages without grammatical final devoicing should have fewer voiced codas than voiced onsets. This rigor is difficult to achieve without detailed phonemic lexicons.

The Crúbadán corpus (Scannell, 2007; cf. Zuraw, 2006) provides word frequency files for thousands of languages, often based on Bible translations and Wikipedia. The Linguistic Data Consortium (LDC) has provided data for many languages in various formats, e.g. conversation transcripts and newswire, from which word frequency files could be easily generated (for a few languages, LDC provides such data directly). An intriguing new source for word frequencies is the Open Subtitles Corpus (Tiedemann, 2009), which collects subtitle data for multiple languages. Therefore, it potentially represents spoken language better than Bible translations or Wikipedia.

There are several challenges in using word lists for research in phonology. First and most obviously, some procedure needs to be applied to translate alphabetic representations to phonemic representations, if such a procedure is possible.[1] But even in cases in which a clear correspondence between the alphabet of a language and its phonemic representation does exist, we may suspect that the data itself is inadequate, or not representative of the phonemic trends of the language. For instance, Daland (2013) discusses *burstiness*, or the possibility that otherwise low-frequency words could bias a sample due to them being over represented in a particular subset of the corpus. A good example of this effect can be found in the Crúbadán entry for Indonesian, in which the word *Indonesia* is the 14th most frequent. This is due to the fact that the word frequencies were created from the Indonesian Wikipedia, a corpus in which the word *Indonesia* is very frequent. For comparison, the word *Indonesia* is not among the 1,000 most frequent words in the word frequency files derived

[*]Corresponding authors

[1]For some questions, using the alphabet directly may be enough (e.g. Piantadosi et al., 2011), but for phonological questions, the use of the alphabet as a proxy for phonemic representations is suspect.

from an Indonesian newspaper collected for Cohen Priva (2017).

Despite burstiness, recent findings suggest that segment frequency, predictability, and informativity values converge to their model values rather quickly (Cohen Priva and Jaeger, 2018), which may follow from the segmental domain being considerably more dense than the word-and-above domain. However, their findings compared subsamples of a corpus to the entire corpus, rather than different corpora to one another. Furthermore, word frequencies were established using spoken corpora. Would it be valid for other studies to rely on word frequency lists from different genres, often less representative of spoken language? An additional limitation is that their findings were based on only one language with millions of word tokens in the entire corpus (the samples were substantially smaller). Our goal in this paper is to assess whether similar findings arise without these limitations, e.g. would Crúbadán-based data be similar to spoken data from the same language, using smaller corpora, and many different languages.

2 Methods and materials

2.1 Word frequency lists

We used word frequencies from three corpora, the Crúbadán Corpus (Scannell, 2007), the Open Subtitles Corpus (Tiedemann, 2009), and conversation transcripts (some of them scripted) from the IARPA Babel program (Adams et al., 2017; Andresen et al., 2019, 2018, 2017; Andrus et al., 2017b; Benowitz et al., 2019; Bills et al., 2015, 2018, 2016; Conners et al., 2016). We only used languages that appeared in the Open Subtitles corpus, or were part of the IARPA Babel program. For every language, we ranked word type by token frequency, only considering words that had the same or more occurrences than the 30,000th ranked word. Additionally, we excluded words that our rules could not translate as well as words whose frequencies in that corpus were lower than 5. Furthermore, we did not use Georgian from the Open Subtitles corpus because we determined that although the words consisted of Georgian script, many were not actually in Georgian, but possibly in Russian.[2] We similarly excluded Haitian Creole from IARPA Babel (Andrus et al., 2017a) because the spelling convention was not consistent with written Haitian Creole. We also excluded words that had any uppercase letters in them in order to discard of irrelevant data, including but not limited to names, acronyms, and companies. The resulting number of types and tokens per corpus are listed in Table 1 for Open Subtitles, and Table 2 for IARPA Babel.

Table 1: Open Subtitles vs. Crúbadán type and token frequencies

Language	Open S. types	Open S. tokens	Crúbadán types	Crúbadán tokens
Bulgarian	23,100	342,000,000	21,300	1,160,000
Catalan	17,700	2,790,000	17,900	1,510,000
Greek	23,100	461,000,000	22,100	1,780,000
Hungarian	29,500	296,000,000	26,300	1,130,000
Indonesian	30,400	75,400,000	14,900	1,690,000
Korean	30,800	5,830,000	28,600	821,000
Malayalam	33,100	1,430,000	14,100	328,000
Tamil	2,950	112,000	28,100	842,000
Tagalog	1,530	68,400	12,700	1,090,000
Turkish	29,000	441,000,000	23,700	795,000

Table 2: IARPA Babel vs. Crúbadán type and token frequencies

Language	Babel types	Babel tokens	Crúbadán types	Crúbadán tokens
Guarani	4,920	391,000	3,150	105,000
Georgian	7,550	408,000	33,900	1,190,000
Swahili	5,240	377,000	16,600	1,680,000
Tamil	9,480	521,000	28,100	842,000
Tagalog	5,370	692,000	12,700	1,090,000
Tok Pisin	1,720	479,000	1,520	1,030,000
Turkish	9,170	663,000	23,700	795,000
Zulu	8,610	416,000	26,900	884,000

2.2 Translation to phonemic representation

For each language in the Open Subtitles and IARPA Babel corpora, we assessed whether it would be possible to translate them to phonemic representations. It is difficult to reconstruct stress reliably, so we did not try to capture this information. We successfully created rules that would translate the following languages (corpus name in parentheses, *o* for open substitles, *b*

[2] For instance, the second most frequent word in Open Subtitles for Georgian is ვ, which (a) does not appear in the Crúbadán Georgian word frequency list and (b) translates to /v/ in Georgian. Therefore, ვ is not a Georgian word but likely the Russian preposition *в*.

for IARPA Babel): Bulgarian (o), Catalan (o), Greek (o), Georgian (b), Guarani (b), Hungarian (o), Indonesian (o), Korean (o), Malayalam (o), Swahili (b), Tagalog (o, b), Tamil (o, b), Tok Pisin (b), Turkish (o, b), and Zulu (b).

The translation procedure involved creating regular expressions that would match letters to their corresponding segments, conditioned by the context in which they were used, with the most specific context taking precedence over less specific contexts. Finally, sporadic string editing operations were used e.g. to treat gemination as a segment followed by a repetition (e.g. /t,ː/), rather than the same segment repeating twice (e.g. /t,t/). The translation procedures were verified against reference translation words for those languages. The full translation procedure, the translation code, and the rules used to translate each language are all available at https://urielcpublic.s3.amazonaws.com/code/SCiL2020Code-2019-09-15.tbz.

2.3 Calculation of information-theoretic properties

We followed standard practice for calculating the information-theoretic measurements (e.g. Aylett and Turk, 2004; van Son and van Santen, 2005; Bell et al., 2009). We calculated three properties. *Segment frequency* is the unigram probability of each segment in the entire corpus, negative $\log_2$ transformed, ignoring types. *Segment type frequency* is the probability of finding each segment in any word type (negative $\log_2$ transformed). *Segment informativity* (Cohen Priva, 2008, 2015) is the expected value of each segment's surprisal (based on maximum-likelihood estimates), using all the preceding phonemes as context (van Son and Pols, 2003). Peripheral segments are likely to be mis-calculated, as they appear in very few word types. Therefore, we removed all segments that occurred more than 50 times less frequently than the most frequent segment (by token). This step is crucial because many alphabets (e.g. Tamil) provide means to represent sounds that are not part of the basic phonemic inventory of the language. The down side is that some non-peripheral phonemes could also be excluded by this procedure. Had we processed American English (for which our translation procedure could not be used, but which does have pronunciation dictionaries), the exclusion criterion would have only led to the exclusion of /ʒ/ and /ɔɪ/. The exclusion of /ʒ/ would have been legitimate, as it is indeed a peripheral phoneme that occurs in restricted contexts, but /ɔɪ/ is not a peripheral phoneme in American English, it is only infrequent.

We also calculated bigram type and token frequency to estimate whether the environments in which segments are found are comparable. These properties are more sparse, thus they are expected to show more bias across corpora (burstiness and per-genre effects are expected). We used add-one smoothing in order to consider all bigrams across corpora.

2.4 Properties of interest

For all five properties, segment type frequency, segment token frequency, segment informativity, bigram type frequency, and bigram frequency, we compare them across corpora. We calculated Pearson correlations between the estimates in one corpus and the estimates of the same properties in the other corpus. We chose Pearson correlations because the values of the different properties are expected to be consistent across corpora, rather than having the same rank. We also report the median difference in bits for the five properties, as the properties are supposed to be near-identical across corpora, not just correlated.

3 Results

3.1 Segment-level properties

In both corpora, all three properties were highly correlated, as shown in Table 3 for Open Subtitles and Crúbadán, and in Table 4 for IARPA Babel and Crúbadán. Correlations were higher overall between the Open Subtitles corpus and Crúbadán than between the IARPA Babel corpora and Crúbadán. Type frequency correlations were higher than token frequency correlations, which means that answering questions such as “how many words have that segment” would be less corpus-dependent than asking “how frequent that segment is.” Figure 1 illustrates the relationship between segment frequencies across the Open Subtitles and Crúbadán, and Figure 2 illustrates the relationship between segment frequencies across IARPA Babel and Crúbadán. Figures 3 and 4 illustrate the relationship of segment informativity between Open Subtitles and Crúbadán, and between IARPA Babel and Crúbadán, respectively. All four figures show that low correlation is usually centered around specific segments rather than all segments. For instance, Tamil /iː/ is a lot more frequent in Open Subtitles than in Crúbadán. This is likely due to the under-representation of the words நீங்கள் and நீ, /niːnkaɭ/ and /niː/ respectively, both of which are second person pronouns, because they are less frequent in written corpora than in spoken corpora (rank 51 and 36, vs. 3 and 13, respectively). Such discrepancies were more likely to affect segments whose type frequencies were low than segments whose type frequencies were high, as verified in a post-hoc correlation test between the absolute difference between the estimates and their type frequency (always positive, statistically significant in 10 out of the 18 comparisons we have).

Table 3: Open Subtitles vs. Crúbadán correlation between information-theoretic properties. For every property, we provide the Pearson *r* correlation, and in parentheses, the median absolute difference in bits.

Language	Seg. type freq.	Seg. token freq	Seg. informativity
Bulgarian	0.99 (0.08)	0.97 (0.13)	0.97 (0.17)
Catalan	1 (0.05)	0.99 (0.12)	0.95 (0.24)
Greek	0.99 (0.06)	0.99 (0.16)	0.92 (0.29)
Hungarian	0.99 (0.07)	0.99 (0.14)	0.98 (0.13)
Indonesian	0.99 (0.13)	0.98 (0.19)	0.98 (0.17)
Korean	0.98 (0.13)	0.98 (0.22)	0.96 (0.17)
Malayalam	0.99 (0.1)	0.98 (0.18)	0.99 (0.11)
Tamil	0.98 (0.17)	0.92 (0.19)	0.83 (0.37)
Tagalog	0.98 (0.29)	0.97 (0.11)	0.92 (0.17)
Turkish	0.99 (0.11)	0.99 (0.13)	0.98 (0.14)

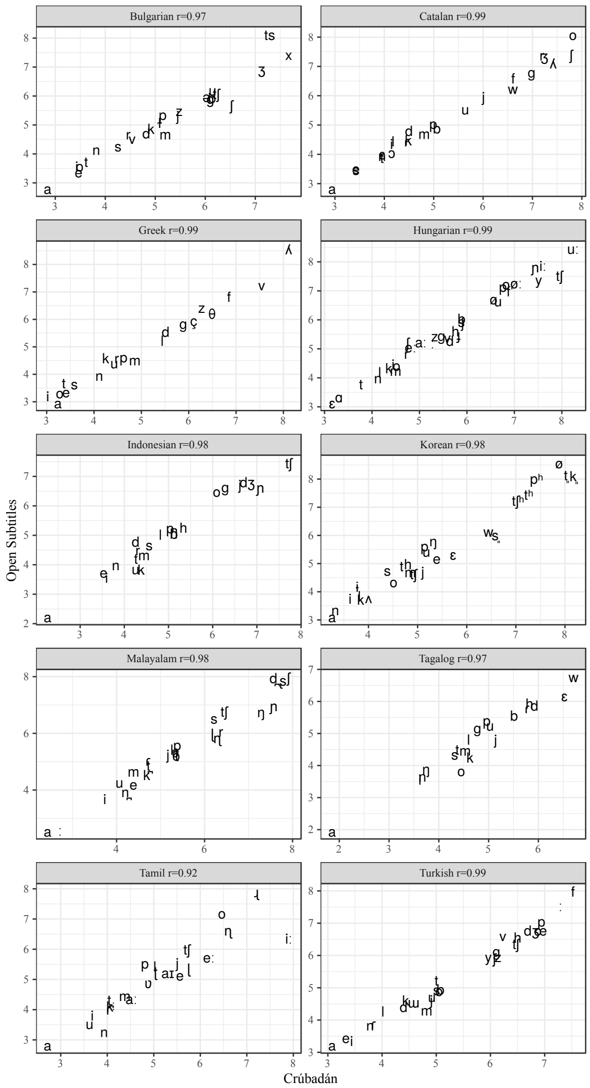


Figure 1: Segment frequency correlation between Open Subtitles and Crúbadán frequency. Both axes are in bits.

3.2 Bigram-level properties

The results are summarized in Table 5 for Open Subtitles and Crúbadán, and in Table 6 for

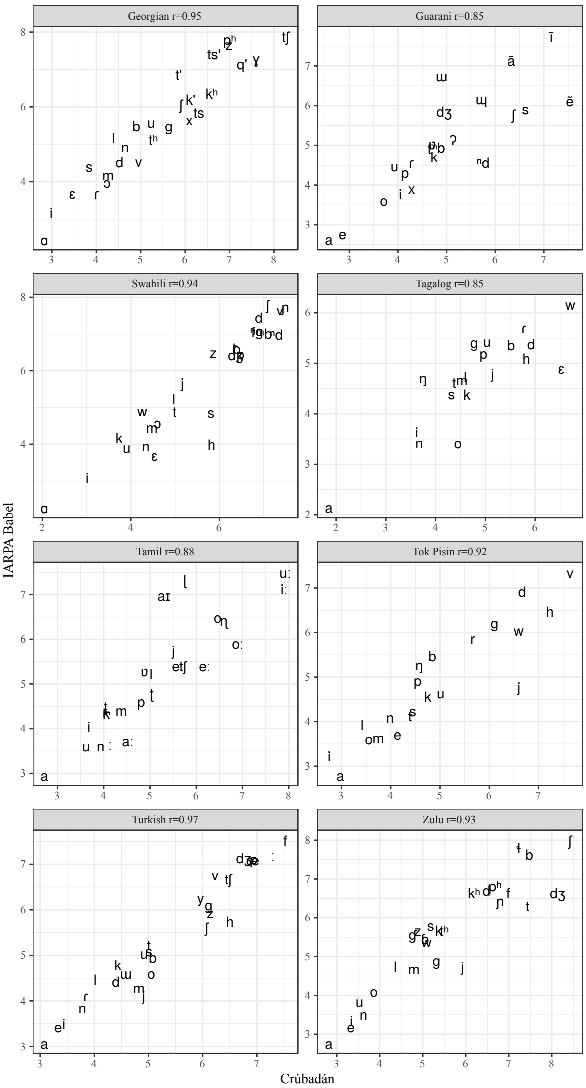


Figure 2: Segment frequency correlation between IARPA Babel and Crúbadán frequency. Both axes are in bits.

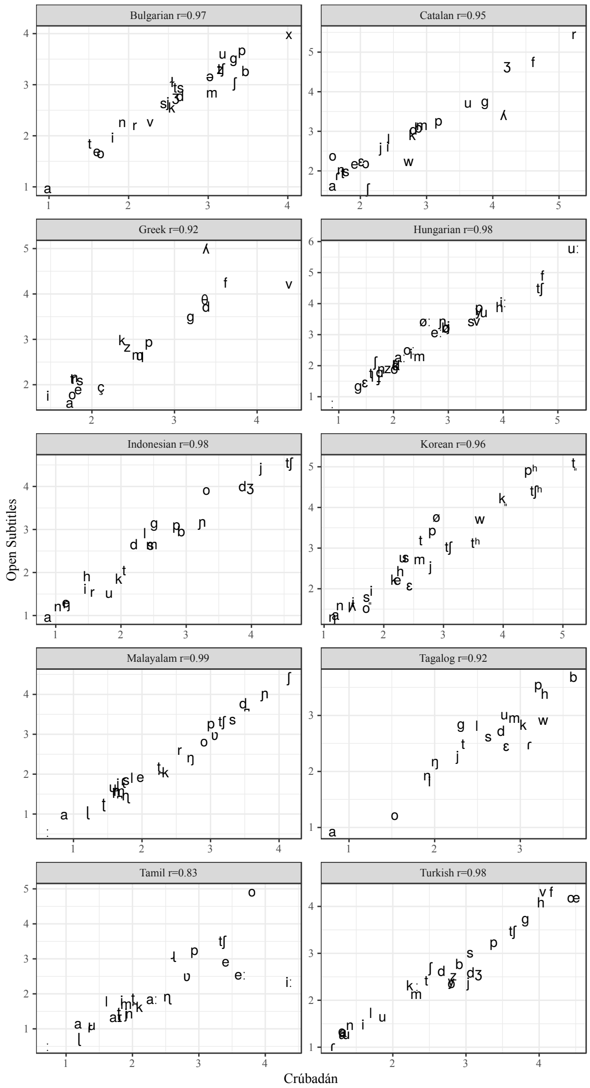


Figure 3: Segment informativity correlation between Open Subtitles and Crúbadán informativity. Both axes are in bits.

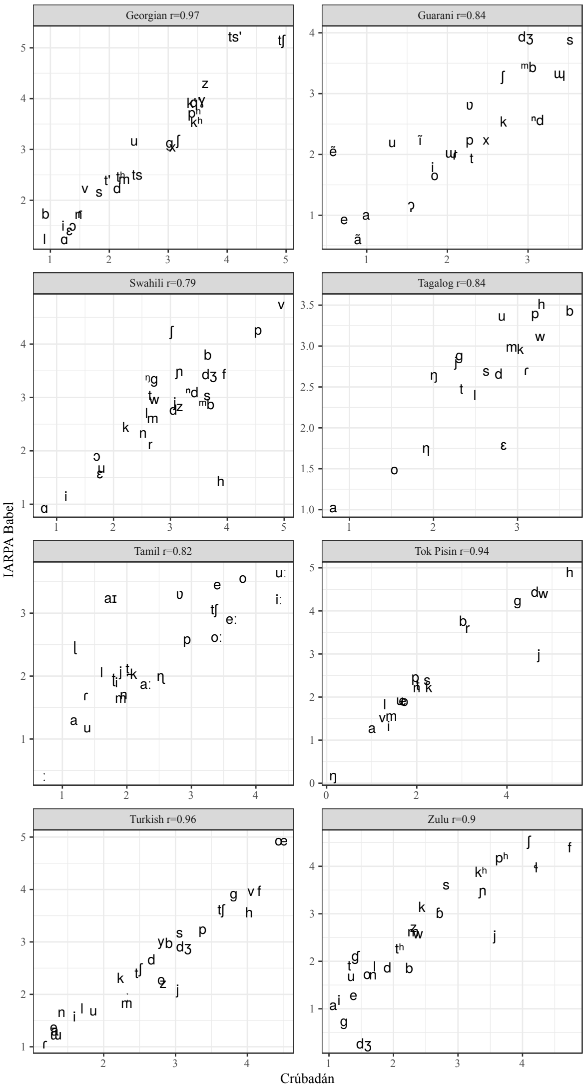


Figure 4: Segment informativity correlation between IARPA Babel and Crúbadán informativity. Both axes are in bits.

Table 4: IARPA Babel vs. Crúbadán correlation between information-theoretic properties. For every property, we provide the Pearson *r* correlation, and in parentheses, the median absolute difference in bits.

Language	Seg. type freq.	Seg. token freq	Seg. informativity
Guarani	0.9 (0.19)	0.85 (0.35)	0.84 (0.32)
Georgian	0.98 (0.22)	0.95 (0.32)	0.97 (0.27)
Swahili	0.96 (0.24)	0.94 (0.27)	0.79 (0.26)
Tamil	0.94 (0.3)	0.88 (0.33)	0.82 (0.3)
Tagalog	0.95 (0.17)	0.85 (0.27)	0.84 (0.19)
Tok Pisin	0.95 (0.22)	0.92 (0.3)	0.94 (0.25)
Turkish	0.99 (0.15)	0.97 (0.18)	0.96 (0.12)
Zulu	0.95 (0.32)	0.93 (0.36)	0.9 (0.34)

IARPA Babel and Crúbadan.

As expected, the correlations were overall lower at the bigram level than at the segmental level, likely due to sparsity issues that we know exist at the word level (Daland, 2013). However, for most languages, the correlations were still impressively high, at Pearson r>.93 and r>.85 for bigram type frequency, representative of Open Subtitles and IARPA Babel's correlations with Crúbadán respectively, and Pearson r>.86 and r>.79 for bigram token frequencies, representative of Open Subtitles and IARPA Babel's correlations with Crúbadán respectively. For reference, assuming that the inherent noise of an experimental population is SD=1 and the sampling noise equals SD=.5, the correlation between test and retest of the same individual is expected to be around Pearson r=.8.

Table 5: Open Subtitles vs. Crúbadán correlation between type and token frequencies for bigrams. For every property, we provide the Pearson *r* correlation, and in parentheses, the median absolute difference in bits.

Language	# bigram types	Bigram type freq.	Bigram token freq
Bulgarian	608	0.98 (0.27)	0.86 (0.56)
Catalan	611	0.97 (0.28)	0.94 (0.58)
Greek	464	0.97 (0.33)	0.88 (0.63)
Hungarian	1202	0.96 (0.39)	0.83 (0.6)
Indonesian	627	0.97 (0.36)	0.91 (0.74)
Korean	705	0.96 (0.45)	0.9 (0.67)
Malayalam	970	0.95 (0.4)	0.9 (0.71)
Tamil	681	0.94 (0.62)	0.9 (0.87)
Tagalog	446	0.93 (0.63)	0.89 (0.74)
Turkish	733	0.97 (0.41)	0.85 (0.63)

Table 6: IARPA Babel vs. Crúbadán correlation between type and token frequencies for bigrams. For every property, we provide the Pearson *r* correlation, and in parentheses, the median absolute difference in bits.

Language	# bigram types	Bigram type freq.	Bigram token freq
Guarani	484	0.85 (0.91)	0.74 (1.68)
Georgian	879	0.93 (0.66)	0.88 (1.29)
Swahili	621	0.91 (0.74)	0.81 (1.12)
Tamil	714	0.9 (0.91)	0.81 (1.47)
Tagalog	479	0.91 (0.48)	0.79 (1.39)
Tok Pisin	357	0.9 (0.56)	0.83 (1.14)
Turkish	724	0.96 (0.43)	0.91 (1.22)
Zulu	729	0.87 (0.81)	0.73 (1.58)

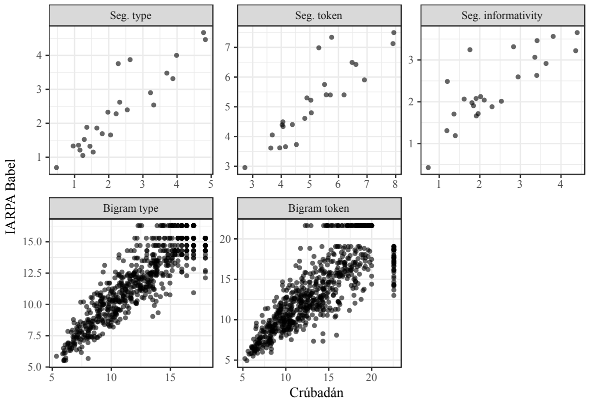


Figure 5: Segment type frequency, token frequency, and informativity, as well as bigram type frequency and bigram informativity for Tamil, by property. Especially for bigram values, it is evident that estimates get progressively worse for low frequency values.

4 Discussion

4.1 Differences across corpora and corpus-usability

We were concerned that the lower correlations between IARPA Babel and Crúbadán, relative to the correlations between Open Subtitles and Crúbadan, were due to the smaller size of the corpus. Cohen Priva and Jaeger (2018) report correlations that approximate >.99 for segment frequency with as few as 100,000 word tokens, a threshold nearly all of our corpora passed (except Open Subtitles for Tagalog). To verify that corpus size is not an issue we ran a post-hoc analysis to predict segment correlations (Fisher-transformed) using log frequencies from the two contributing corpora. Except for a marginal effect for token frequencies in Open Subtitles, there was no correlation. We did observe substantially more interjections, false-starts, loanwords, and conversation-starting / ending in IARPA Babel than in either Crúbadán or Open Subtitles, which is to be expected given the type of the corpus. We are not sure why different languages show this effect to different extents, but given the number of comparisons we have, it would seem that the lower boundary on within-language correlations is still high enough to support the study of phonological properties using corpora of different types and with relatively high degrees of noise.

4.2 Reducing noise

Given that some degree of noise does exist when switching corpus types, it is important to ask what could be done to decrease the amount of noise. One parameter researchers can control is reliance on low-frequency segments and bigrams as well as the use of more robust statistics.

Certainty of information-theoretic values diminishes for less frequent segments and bigrams, which are more easily swayed by word-level frequency effects. Figure 5 shows the correlations for Tamil. It is evident that the estimates for lower-frequency bigrams (and to some extent, individual segments) are worse than for high-frequency segments. Studies that cannot tolerate the lower-precision that is associated with changes across genres could therefore focus on high-frequency segments and contexts.

5 Conclusion

We checked whether segment type frequency, segment token frequency, segment informativity, as well as bigram type and token frequency could be reliably estimated across different corpus types genres. We showed that segments were more reliably estimated than bigrams and that type frequencies were more reliably estimated than token frequencies. However, even for the least similar corpora, Crúbadán and

IARPA Babel, the reliability of measurements was substantial, and likely not larger than for many experimental designs. We also found that high-frequency elements were more reliably estimated than lower-frequency ones. We therefore believe that corpus-based research in phonology can mitigate the concerns related to generalizations across genre and corpus types.

6 Acknowledgements

The work presented in this paper was supported by NSF Grant, awarded to the first author, BCS-1829290. Many thanks go out to Justin Bai (Indonesian, Korean, Tok Pisin, Turkish), Abi Creighton (Catalan, Tagalog, Zulu), Maddie Critz (Georgian), Delphine Morse Mahos (Swahili), Becky Mathew (Indonesian, Korean, Malayalam, Georgian, Tamil), and Bill Mizgerd (Bulgarian, Guarani) for creating the transcription rules necessary for this research to be conducted. The languages each individual was responsible for are indicated in parentheses. We would also like to thank Robert Daland and the two anonymous reviewers for providing constructive feedback on this research.

References

Nikki Adams, Aric Bills, Thomas Conners, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Willa Lin, Jennifer Melot, Jessica Ray, Anton Rytting, Wade Shen, Ronnie Silber, Evelyne Tzoukermann, and Jamie Wong. 2017. IARPA Babel Zulu Language Pack IARPA-babel206b-v0.1e LDC2017s19. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Jess Andresen, Aric Bills, Thomas Conners, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Kirill Kozlov, Nicolas Malyska, Jennifer Melot, Michelle Morrison, Josh Phillips, Jessica Ray, Anton Rytting, Wade Shen, Ronnie Silber, Evelyne Tzoukermann, and Jamie Wong. 2017. IARPA Babel Swahili Language Pack IARPA-babel202b-v1.0d. LDC2017s05. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Lucy Andresen, Aric Bills, Claudia Brugman, Thomas Conners, Anne David, Eyal Dubinski, Jonathan G Fiscus, Ketty Gann, Mary Harper, Michael Kazi, Hanh Le, Nicolas Malyska, Arlene Maurillo, Jennifer Melot, Shelley Paget, Jane Elizabeth Prebble, Jessica Ray, Fred Richardson, Anton Rytting, and Sinney Shen. 2019. IARPA Babel Guarani language pack IARPA-babel305b-v1.0c LDC2019s08. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Lucy Andresen, Aric Bills, Thomas Conners, Luanne Dela Cruz, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Hanh Le, Arlene Maurillo, Jennifer Melot, Josh Phillips, Jessica Ray, Anton Rytting, Wade Shen, Ronnie Silber, and Evelyne Tzoukermann. 2018. IARPA Babel Cebuano Language Pack IARPA-babel301b-v2.0b LDC2018s07. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Tony Andrus, Aric Bills, Thomas Conners, Erin Smith Crabb, Eyal Dubinski, Jonathan G Fiscus, Breanna Gillies, Mary Harper, T.J Hazen, Brook Hefright, Amy Jarrett, Hanh Le, Jessica Ray, Anton Rytting, Wade Shen, Ronnie Silber, and Evelyne Tzoukermann. 2017a. IARPA Babel Haitian Creole Language Pack IARPA-babel201b-v0.2b LDC2017s03. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Tony Andrus, Aric Bills, Miriam Corris, Eyal Dubinski, Jonathan G Fiscus, Breanna Gillies, Mary Harper, T.J Hazen, Brook Hefright, Amy Jarrett, Hanh Le, Jessica Ray, Anton Rytting, Ronnie Silber, Wade Shen, and Evelyne Tzoukermann. 2017b. IARPA Babel Vietnamese Language Pack IARPA-babel107b-v0.7. LDC2017s01. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Arto Anttila. 1997. Deriving variation from grammar: a study of Finnish genitives. In Frans Hinskens, Roeland van Hout, and Leo Wetzels, editors, *Variation, change and phonological theory*, pages 35–68. John Benjamins, Amsterdam.

Matthew Aylett and Alice Turk. 2004. The smooth signal redundancy hypothesis: a functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, 47(1):31–56.

Alan Bell, Jason Brenier, Michelle Gregory, Cynthia Girand, and Daniel Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.

Daniel Benowitz, Aric Bills, Thomas Conners, Anne David, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Brook Hefright, Hanh Le, Jennifer Melot, Jessica Ray, Anton Rytting, Wade Shen, Rosanna Smith, Wade Shen, Ronnie Silber, and Evelyne Tzoukermann. 2019. IARPA Babel Lithuanian Language Pack IARPA-babel304b-v1.0b. LDC2019s03. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Aric Bills, Thomas Conners, Miriam Corris, Anne David, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Alice Kaiser-Schatzlein, Jennifer Melot, Shelley Paget, Jessica Ray, Anton Rytting, Wade Shen, Ronnie Silber, Evelyne Tzoukermann, and Arun Viswanath. 2015. IARPA Babel Tamil Language Pack IARPA-babel204b-v1.1b. LDC2017s13. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Aric Bills, Thomas Conners, Anne David, Eyal Dubinski, Jonathan G Fiscus, Mary Harper, Brook Hefright, Kirill Kozlov, Jennifer Melot, Jessica Ray, Anton Rytting, Josh Phillips, Marle Walter, Wade Shen, Ronnie Silber, and Evelyne Tzoukermann. 2018. IARPA Babel Kazakh Language Pack IARPA-babel302b-v1.0a. LDC2018s13. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Aric Bills, Anne David, Eyal Dubinski, Jonathan G Fiscus, Simon Hammond, Ketty Gann, Mary Harper, Brook Hefright, Michael Kazi, Julie Lam, Jessica Ray, Fred Richardson, Anton Rytting, and Marle Walter. 2016. IARPA Babel Georgian Language Pack IARPA-babel404b-v1.0a LDC2016s12. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Paul Boersma. 1998. *Functional Phonology*. Ph.D. thesis, University of Amsterdam.

Uriel Cohen Priva. 2008. Using information content to predict phone deletion. In *Proceedings of the 27th West Coast Conference on Formal Linguistics*, pages 90–98, Somerville, MA. Cascadilla Proceedings Project.

Uriel Cohen Priva. 2015. Informativity affects consonant duration and deletion rates. *Laboratory Phonology*, 6(2):243–278.

Uriel Cohen Priva. 2017. Informativity and the actuation of lenition. *Language*, 93(3):569–597. (preprint).

Uriel Cohen Priva and T. Florian Jaeger. 2018. The interdependence of frequency, predictability, and informativity. *Linguistics Vanguard*, 4.

Thomas Conners, Jonathan G. Fiscus, Breanna Gillies, Mary Harper, T. J. Hazen, Amy Jarrett, Willa Lin, María Encarnación Pérez Molina, Shawna Rafalko, Jessica Ray, Anton Rytting, Wade Shen, and Evelyne Tzoukermann. 2016. IARPA Babel Tagalog Language Pack IARPA-babel106-v0.2g. LDC2016s13. Technical report, Linguistic Data Consortium, Philadelphia, PA.

Robert Daland. 2013. Variation in the input: a case study of manner class frequencies. *Journal of Child Language*, 40:1091–1122.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm workshop on variation within Optimality Theory*, pages 111–120.

John J. McCarthy. 1986. OCP effects: Gemination and antigemination. *Linguistic Inquiry*, 17(2):207–263.

Steven T. Piantadosi, Harry J Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*.

Kevin P Scannell. 2007. The Crúbadán project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, volume 4, pages 5–15.

R. J. J. H. van Son and L. C. W. Pols. 2003. How efficient is speech? *Proceedings of the Institute of Phonetic Sciences*, 25:171–184.

R.J.J.H. van Son and J.P.H van Santen. 2005. Duration and spectral balance of intervocalic consonants: a case for efficient communication. *Speech Communication*, 47:100–123.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Kie Zuraw. 2006. Using the web as a phonological corpus: A case study from Tagalog. In *Proceedings of the 2Nd International Workshop on Web As Corpus*, WAC '06, pages 59–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

Modeling Behavior in Truth Value Judgment Task Experiments

Brandon Waldon
Stanford University
bwaldon@stanford.edu

Judith Degen
Stanford University
jdegen@stanford.edu

Abstract

Truth Value Judgment Task experiments (TVJTs) are a common means of investigating pragmatic competence, particularly with regards to scalar inference. We present a novel quantitative linking function from pragmatic competence to participant behavior on TVJTs, based upon a Bayesian probabilistic model of linguistic production. Our model captures a range of observed phenomena on TVJTs, including intermediate responses on a non-binary scale, population and individual-level variation, participant endorsement of false utterances, and variation in response due to so-called scalar diversity.

1 Introduction

In Truth Value Judgment Task experiments (TJVTs), participants are asked whether a given sentence is, e.g., 'right' or 'wrong' (or 'true' or 'false', etc.), often in a context of evaluation. In the field of experimental pragmatics, participant judgments in TVJT paradigms have been particularly important for investigating pragmatic competence, especially as it relates to scalar implicature (Noveck, 2001; Noveck and Posada, 2003; Bott and Noveck, 2004; De Neys and Schaeken, 2007; Geurts and Pouscoulous, 2009; Chemla and Spector, 2011; Degen and Goodman, 2014; Degen and Tanenhaus, 2015). On the traditional view of pragmatic competence and its link to TVJT responses, scalar implicature is assumed - following Grice (1975) - to be a binary and categorical phenomenon, in the sense that a given utterance is assumed to categorically either give rise to an implicature or not, depending on contextual, cognitive, and linguistic factors. To experimentalists operating on this assumption, a participant's judgment on a particular trial in a TVJT reflects whether or not a scalar implicature was computed in context.

For example, a 'wrong' judgment of the sentence *John ate pizza or a sandwich*, in a context in which the stronger utterance alternative *John ate pizza and a sandwich* is true and equally relevant, is typically interpreted as a "pragmatic" judgment: participants must have recognized that in such a context, the *or*-sentence is true yet underinformative. Pragmatically enriching it to *John didn't eat both pizza and a sandwich* via scalar inference makes it contextually false. Conversely, an answer of 'right' on this view reflects a "literal" semantic interpretation whereby the implicature is not computed (i.e. *John ate pizza or a sandwich - and possibly both*).

This linking assumption underpins the vast majority of TVJT literature relating to scalar inference (Noveck, 2001; Papafragou and Musolino, 2003; Geurts and Pouscoulous, 2009; Doran et al., 2012; Potts et al., 2015). In an early example, Papafragou and Musolino (2003) observe that children accept true but underinformative sentences in a TVJT at a relatively high rate relative to adults, and that this rate is modulated by the particular linguistic scale invoked on a given trial of the experiment (i.e. *some/all* vs. *finish/start* vs. cardinal numbers). The authors argue from this result that scalar implicature computation is dependent upon linguistic scale as well as on a child's recognition of the communicative goals of her interlocutor.

Though widely employed, this linking assumption for TVJTs is associated with a host of problems discussed by Jasbi et al. (2019). Following those authors as well as Tanenhaus (2004), we take these open problems to be indicative of a larger issue in linguistics, namely that the linking hypotheses which bridge linguistic theory and experimentally-elicited behavior are often underdeveloped, underspecified, or (in some cases) absent in experimental studies. In the service of providing a proof of concept for how this issue may be addressed by future researchers, we propose and evaluate a novel account of participant response in TVJT paradigms based on an explicit and quantitatively specified linking function rooted in a probabilistic theory of pragmatic competence. The general idea is that participants' responses in TVJT experiments are related to the probability with which a cooperative pragmatic speaker would have produced the observed utterance (e.g., *John ate pizza or a sandwich*) in order to communicate the meaning presented to participants as fact (e.g., that John ate both pizza and a sandwich). This probabilistic *production* based view departs substantially from the previous widespread assumption that truth-value judgments are a measure of *interpretation*.

Before turning to the specifics of the account, we briefly review some of the open problems in the TVJT literature that motivate the re-thinking of linking functions in TVJT paradigms:

Intermediate judgments: When provided more than two response options in a TVJT, a sizable proportion of participants rates underinformative sentences using the intermediate response options - for example, as only 'kind of [right / wrong]', or 'neither [right nor wrong]'. Katsos and Bishop (2011), for example, provided participants with three response options and observed substantial selection of the intermediate option. They interpreted the choice of this intermediate option as being the result of the computation of an implicature, but a priori, there is no reason to favor this linking assumption over one whereby the intermediate response is associated with a literal semantic interpretation. More generally, it is not clear how the outputs of a binary model of scalar implicature (i.e. implicature or ¬implicature) should relate to non-binary responses on TVJTs.

Population-level variation: In order to explain behavioral variation in contexts where one expects a scalar inference, an adherent to the categorical view of scalar implicature must stipulate that a) not all participants calculated the implicature; or b) some participants who calculated the implicature showed divergent behavior due to some independent mechanism which masked the 'correct' implicature behavior; or some combination of (a) and (b). However, and despite the prevalence of variation at the population level in reported TVJT experiments, even a qualitative analysis of this kind of variation is largely absent from the experimental scalar implicature literature.

Scalar diversity: Doran et al. (2012), following Papafragou and Musolino (2003) inter alia, report that judgments of true but underinformative sentences vary according to the particular linguistic material contained within the sentence, in particular the relevant linguistic scale. They conclude that variation among scalar implicatures is a function of the scale itself (see also van Tiel et al. 2014 for further support for scale-based scalar diversity in a non-TVJT paradigm).

Whether this variation is truly due to inherent features of the linguistic scale (or, e.g., prior world knowledge, or other linguistic material, or other confounding features of the experimental context) is an open question which warrants investigation beyond the scope of this paper. Below, we analyze data from a TVJT where different rates of exhaustive interpretation were observed between a putative lexical scale (<*and, or*>) and a putative ad-hoc, context-dependent pragmatic scale. Our analysis of the data suggests that in this instance, (at least some) variation at the level of linguistic scale may be reduced to more general aspects of pragmatic competence.

Endorsement of false utterances: Invariably, a proportion of participants in TVJTs accepts strictly false sentences. For example, in the study we analyze below, a substantial number of participants rated conjunctions $A \wedge B$ as partially correct in contexts where only A was true. The most common approaches to this type of data are either to use it as the basis of an exclusion criterion or to simply consider it meaningless noise. Doran et al. (2012), for example, exclude participants whose performance deviates by more than two standard deviations from the mean response on 'control' sentences whose semantic contents are consistent with the context of evaluation (and which do not admit of potentially contradictory pragmatic enrichments) or whose semantic contents contradict the context. Katsos and Bishop (2011) report that 2.5% of false sentences in their experiment were endorsed by child participants. On the standard linking assumption, these data are difficult to make sense of, but we will show that they are within the scope of a satisfactory analysis of TVJT behavior.

The remainder of the paper is structured as follows: in Section 2, we summarize the results

Condition	Response Options
Binary	'Right', 'Wrong'
Ternary	'Right', 'Neither', 'Wrong'
Quaternary	'Right', 'Kinda Right', 'Kinda Wrong','Wrong'
Quinary	'Right', 'Kinda Right', 'Neither', 'Kinda Wrong', 'Wrong'

Table 1: Response-option conditions of Jasbi et al. (2019)'s TVJT study.

of a recently reported TVJT study that exemplifies the features discussed above: intermediate judgments, population-level variation, scalar diversity, and participant endorsement of false utterances. Section 3 presents our novel quantitative model of the data from that study. Building on insights from the Bayesian probabilistic literature on pragmatic competence (Frank and Goodman, 2012; Goodman and Stuhlmüller, 2013), we model participants as making judgments about a soft-optimal pragmatic speaker whose production choices are a function of utterances' contextual informativeness. On our analysis, participants furthermore expect that the speaker sometimes produce strictly false utterances that are nonetheless somewhat contextually useful. We show that this analysis provides broader empirical coverage over the traditional assumptions discussed above.[1]

2 TVJT Data

2.1 Experiment Materials, Design and Procedure

Jasbi et al. (2019) report the results of a TVJT designed to test whether linking hypothesis and number of response options modulate the researcher's inferences about scalar implicature rates. In their study, number of response options varied between two and five as a between-subjects manipulation. Conditions are summarized in Figure 1. Participants (n = 200) were first shown six cards (Table 2) featuring one or two of the following animals: a cat, a dog, and an elephant. On every trial, participants saw one of the six cards, and a blindfolded cartoon character Bob made guesses as to what animals were on the card. Participants were asked to rate Bob's guesses using the response options available in their particular condition.

Bob made the following guess types: simple declaratives (e.g., *There is a cat*), conjunctions (e.g., *There is a cat and a dog*), and disjunctions

[1]Data and code for all analyses and graphs are available at `http://github.com/bwaldon/tvjt_linking`.

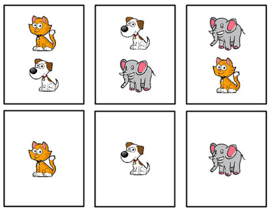


Table 2: Cards used in Jasbi et al. (2019)'s TVJT.

(e.g., *There is a cat or a dog*). Card types were crossed with guess types in this study such that a card containing an animal *X* could be presented with a guess of *There is an X*, *There is an X or a Y* (where *Y* is some animal distinct from *X*), *There is an X and a Y*, or *There is a Y*; cards containing two animals *X* and *Y* could be presented with a guess of *There is an X*, *There is an X or a Y*, *There is an X and a Y*, or *There is a Z* (where *Z* is some animal distinct from *X* and *Y*).

The researchers elicited 3 judgments per participant for each combination of card and guess type.

2.2 Results and Discussion

Proportions of responses for each card-guess type in each response-option condition are shown in Figure 1, with rows presenting behavior aggregated across one and two-card conditions.

The results of the study illustrate the several open empirical issues associated with TVJTs more generally. First, participants routinely reported **intermediate judgments** between 'Right' and 'Wrong' in those conditions where intermediate response options were available. In the Quaternary and Quinary response-option conditions, for example, the intermediate judgment of 'Kinda Right' was the single most-selected response option in two-animal card conditions where Bob's guess was true but underinformative (i.e. either a simple delcarative or a disjunction).

The results also exemplify the issue of **population-level variation**: for example, although behavioral patterns are otherwise fairly categorical in the Binary condition, participant judgments were roughly split between 'Right' and 'Wrong' for underinformative uses of disjunction on two-animal card conditions. A visual inspection of the results suggests even more variation in the population as number of response options increase. The authors furthermore reported **individual-level variation**: qualitatively similar trials (e.g. two trials involving underinformative disjunction) sometimes received different re-

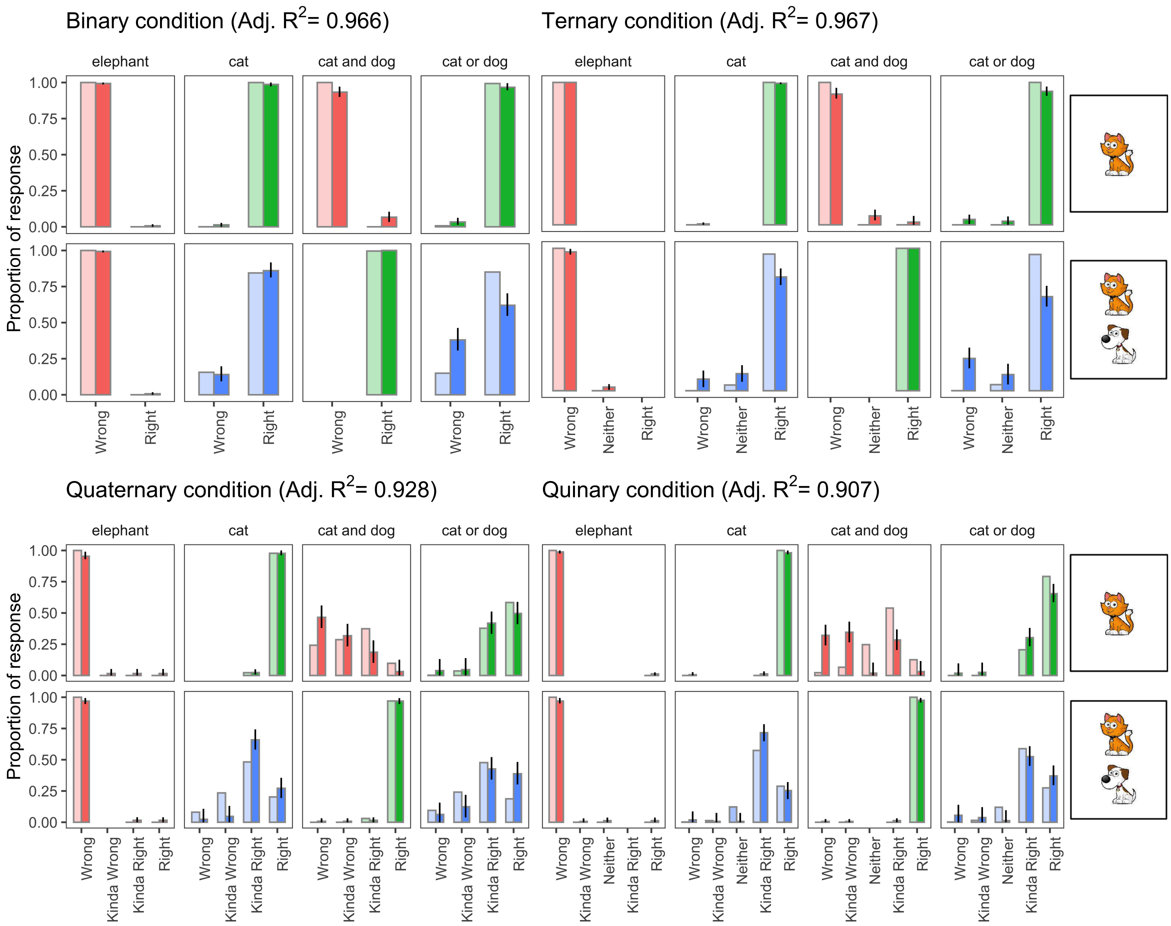


Figure 1: Model predictions (light bars) plotted against empirical results (dark bars) from Jasbi et al.'s (2019) TVJT study. Error bars indicate 95% multinomial confidence intervals. Red and green bars indicate false and true trials, respectively; blue bars indicate implicature trials.

sponses from the same participant.

Comparison of judgments of true but underinformative simple declaratives (i.e. *There is an X*) to judgments of true but underinformative disjunctions (i.e. *There is an X or a Y*) on two-animal card conditions revealed some amount of **scalar diversity**. Following Horn (1972), exposure to the disjunctive connective *or* canonically activates an informationally-stronger scalemate *and* as a pragmatic alternative to give rise to an exclusive interpretation. In contrast, the pragmatic scale in the case of the simple declarative is constructed in a more context-dependent manner. To illustrate, in a two-animal card context where the card features both a cat and a dog, the listener considers a partially-ordered pragmatic scale of *cat and dog*, *cat*, and *dog*, where the conjunction outranks its scalemates in terms of informational strength. Thus, an utterance of *cat* activates *cat and dog* as an alternative to give rise to an the exhaustive interpretation (*There is only a cat on the card*).

In the Binary and Ternary conditions, underinformative uses of *or* resulted in substantially higher rates of 'Wrong' responses than did underinformative simple declaratives, suggesting that at the population level, *or* was interpreted more exhaustively than the simple declarative. However, this pattern was reversed in the Quaternary and Quinary conditions, in which underinformative simple declaratives were more likely to be considered only 'Kinda Right' and less likely to be considered simply 'Right' compared to underinformative disjunctions. This pattern suggests that in the Quaternary and Quinary conditions, simple declaratives were interpreted more exhaustively than disjunctions.

Finally, the data in the Quaternary and Quinary conditions also reveal substantial **participant endorsement of false utterances**. Note specifically that in one-animal card trials, the conjunctive guess (e.g. *cat and dog*) is strictly false; thus, we might naïvely expect a priori that participants categorically judge these utterances to be 'Wrong' in all conditions. Yet when given the option to rate this sentence 'Kinda Right' or 'Kinda Wrong', participants often did so. In all other conditions where the utterance was strictly false (e.g. a guess of *elephant* for a card containing a cat or a cat and dog), behavior was effectively categorical. That is, rates of endorsement of false utterances varied according to the particular way in which the sentence was false in context.

In sum, the data collected by Jasbi et al. (2019) reflect a range of behavioral patterns unaccounted for by the traditional categorical view of scalar inference and corresponding standard linking assumptions. Below, we report an analysis of their data that aims to predict these phenomena.

3 Analysis

3.1 Cognitive model

Our analysis implements a proposal outlined by Jasbi et al. (2019), couched in the Rational Speech Act (RSA) framework (Frank and Goodman, 2012; Goodman and Stuhlmüller, 2013). RSA provides a Bayesian, probabilistic account of pragmatic competence. In RSA, the pragmatic inferences drawn by listeners are represented as probability distributions over meanings which the speaker plausibly intended to convey with a given observed utterance. The probability of this listener (L_1) attributing an intended meaning m to a speaker who produces an utterance u is calculated from a prior probability distribution over potential world states P_w as well as from L_1's expectations about the linguistic behavior of the speaker S_1.

$$P_{L_1}(m|u) \propto P_{S_1}(u|m) \cdot P_w(m)$$

P_{S_1} is modeled as a probability distribution over possible utterances given the speaker's communicative intentions m. This speaker produces utterances that soft-maximize utility, where utility is defined via a tradeoff between an utterance's cost C and its contextual informativeness, calculated from the representation of a literal listener L_0 whose interpretation of an utterance u is in turn a function of the truth conditional meaning $[[u]](m)$ and of her prior expectations $P_w(m)$ regarding the likelihood of possible world states. The extent to which the speaker maximizes utility is modulated by a parameter α – the greater α, the more the speaker produces utterances that maximize utility.

$$P_{S_1}(u|m) \propto e^{\alpha(\ln L_0(m|u) - C(u))}$$

$$P_{L_0}(m|u) \propto [[u]](m) \cdot P_w(m)$$

In RSA (and contra the traditional view), pragmatic inferences are not categorical computations of enriched meanings over the semantic denotations of utterances. For example, exclusive interpretations of *or* are represented in RSA as a positive shift in the posterior probability of an exclusive meaning, relative to its prior probability.

In other words, 'implicature' is not a theoretical construct in the RSA framework, absent additional stipulations regarding how to go from probability distributions to binary, categorical inferences. This is an advantage: providing a probabilistic representation of both the speaker's utterance choices and the listener's resulting posterior beliefs after observing an utterance puts us one step closer to accounting for the quantitative behavioral patterns observed in tasks such as TVJTs.

3.2 Behavioral model

Jasbi et al. (2019) proposed but did not systematically test a simple linking hypothesis: rather than providing one response if an implicature is computed and another if it isn't, a participant in a TVJT experiment provides a particular response to an utterance u if the probability of u given a meaning represented by m lies within a particular probability interval on the distribution $P_{S_1}(u|m)$.[2] The participant is modeled as a responder R, who in a binary forced-choice task between 'Right' and 'Wrong' responds 'Right' to an utterance u in world m just in case $P_{S_1}(u|m)$ meets or exceeds some probability threshold θ:

$$R(u, m, \theta) = \begin{cases} \text{'Right'} & \text{iff } P_{S_1}(u|m) \geq \theta \\ \text{'Wrong'} & \text{otherwise} \end{cases}$$

The model is extended straightforwardly to an experiment in which participants have a third response option (e.g. 'Neither'), as in the Ternary condition. In this case, the model specifies two probability thresholds: θ_1, the minimum standard for an utterance in a given world state to count as 'Right', and θ_2, the minimum standard for 'Neither'. Thus, in the Ternary condition:

$$R(u, m, \theta) = \begin{cases} \text{'Right'} & \text{iff } P_{S_1}(u|m) \geq \theta_1 \\ \text{'Neither'} & \text{iff } \theta_1 > P_{S_1}(u|m) \geq \theta_2 \\ \text{'Wrong'} & \text{otherwise} \end{cases}$$

Applying a similar logic allows for the specification of linking hypotheses for TVJTs with an

[2] Following Degen and Goodman (2014), the authors argue that conceptually, behavior on TVJTs is better modeled as a function of an agent's representation of a pragmatic speaker rather than of a pragmatic listener.

arbitrary number of response options.

The intuition behind the threshold model is as follows: participants should disprefer utterances that are relatively unexpected. Thus, high S_1 production probability for a given utterance in context makes it more likely that the utterance receives a positive evaluation in the TVJT – expressed by ordered response options above 'Wrong'. Conversely, the more unexpected an utterance is, the more likely it is to be judged as 'Wrong'. Underinformative utterances of the sort that have traditionally been used to assess 'implicature rates' are precisely the kinds of utterances that are unexpected from informative speakers and are therefore likely to be rated as 'Wrong'.

Here, we assess the quality of this linking hypothesis on the dataset from Jasbi et al. (2019). To that end, we first specify the space of possible meanings and utterances that inform a participant's pragmatic competence in this task. We assume that participants have uniform prior expectations of seeing any of the six possible cards in the experiment. We further assume that participants have uniform prior expectations of a speaker producing any of the four utterance types with which a card may have been crossed. For example, if the card featured either just a cat or both a cat and a dog, we represent the participant as having uniform prior expectations of a speaker producing the guesses *elephant*, *cat*, *dog*, *cat and dog*, or *cat or dog* (that is, we do not posit a cost asymmetry between possible utterances).[3]

For illustrative purposes, the 'Simple Bayesian' bars in Figure 2 display marginal distributions over possible utterances produced by S_1 given these assumptions for the utterance and meanings priors, as well as an arbitary value of 1 for the optimality parameter α, and given that the speaker intends either to communicate the meaning that (just) a cat is on the card or that both a cat and a dog are. The speaker distributions reveal two conceptual issues for the threshold response model proposed by Jasbi et al (2019).

First, the probability of S_1 producing the strictly false guess of *cat and dog* should be zero if the card contains just a cat. This is because the literal listener probability P_{L_0} of inferring the 'only cat' meaning given *cat and dog* is zero by virtue of the fact that the utterance is strictly false in this world state. Thus, any model of response that is a function of P_{S_1} as specified predicts that participants categorically rate the *cat and dog* guess as 'Wrong' in this context, contrary to what is observed in the Quaternary and Quinary conditions.

Second, the probability of producing disjunctions is lower than the probability of producing simple declarative guesses in two-animal card contexts. This asymmetry is advantageous in the case of the Binary and Ternary response data: assuming a threshold for 'Right' positioned between $P_{S_1}(\textit{cat or dog}|\text{cat and dog})$ and $P_{S_1}(\textit{cat}|\text{cat and dog})$, we predict correctly that underinformative simple declaratives should be judged 'Right' more often than underinformative disjunctions. But the asymmetry in S_1 probabilities therefore predicts the wrong pattern of responses on corresponding trials in the Quaternary and Quinary conditions.

We argue that these two seemingly disparate issues can be mediated by a common solution. In particular, we propose a revision to the simple Bayesian inference story above, whereby pragmatically-competent listeners either expect speaker productions as directly sampled from the P_{S_1} distribution, <u>or</u> that those utterance production probabilities inform a second conditional probability distribution of utterances given utterances, the 'Partial Truth' utterance distribution $P_{S_{PT}}$:

$$P_{S_{PT}}(u'|u) \propto \sum_{m \in [\![u]\!]} P_{S_1}(u'|m)$$[4]

The 'Partial Truth' distribution is a generalized way of modeling a speaker who makes assertions that are sometimes strictly false in light of her intended meaning. Recall that the semantic content of any possible utterance choice made by S_1 is a set of possible worlds and is therefore consistent with meanings unintended by the speaker. S_{PT} models the speaker's soft-optimal production probabilities given these unintended meanings, renormalizing the pragmatic speaker's production probabilities over all possible worlds consistent with utterance choices sampled from P_{S_1}.

[3]We include *dog* as a possible guess because we posit that participants have no reason a priori to expect the other true and underinformative simple declarative - *cat* - over this equally informative guess in two-animal card conditions.

[4]For our implementation of S_{PT}, we restrict the distribution such that u' must entail (or be entailed by) u in order to have probability above 0. Without this restriction, S_{PT} could in principle assign high probability to utterances which have no relevance to the question under discussion (i.e. "What animals are on the card?"), by virtue of those utterances' assertability in worlds consistent with u. A systematic exploration of the linguistic alternatives available to S_1 (as well as S_{PT}) is a question we must leave to future work.

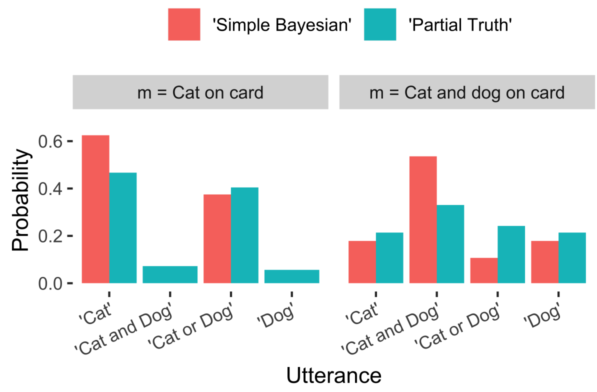


Figure 2: Simulated S_1 production probabilities.

To illustrate: suppose a speaker intends to communicate that many (but not all) of the X are Y, and has quantifier choices *many* and *all*. The only possible utterance choice for the simple Bayesian S_1 speaker is *many*, which is semantically consistent with the intended meaning. But the lower-bounded quantifier *many* is also semantically consistent with an 'all of the X are Y' meaning, which in turn is consistent with the utterance choice *all*. By S_{PT}, we have some nonzero expectation that the speaker will use *all* to communicate the 'many (but not all) of the X are Y' meaning.[5] Thus, a pragmatic listener who hears *all* from the 'Partial Truth' speaker will have a nonzero expectation that *all* should receive an imprecise, non-maximal interpretation. In other words, S_{PT} provides a generalized way of formalizing 'loose-talk' production behavior (Lasersohn, 1999).[6]

The 'Partial Truth' bars in Figure 2 visualize marginal distributions over utterances given an arbitrary 0.6 probability that the speaker samples from the $P_{S_{PT}}$ distribution after sampling from P_{S_1}. The 'Partial Truth' speaker assigns nonzero probability to a guess of *cat and dog* even when the speaker's intended meaning is the single-animal cat card, largely due to the fact that the optimal guess in this context (*cat*) is truth-conditionally consistent with a two-animal card that makes *cat and dog* both true and pragmatically optimal.[7] Moreover, this speaker assigns greater probability to a guess of *cat or dog* in two-animal contexts and down-weights the probability of producing simply *cat*: the optimal utterance in this context (*cat and dog*) is consistent with several world states in which the disjunction *cat or dog* is assertable and with relativley fewer worlds in which *cat* is assertable.

3.3 Quantitative model evaluation

We now turn to a quantitative assessment of the threshold model of response, having addressed two ways in which the unenriched S_1 representation would fail to qualitatively capture behavioral patterns in Jasbi et al (2019)'s TVJT study. Additionally, following Jasbi et al., we recognize that if threshold values were made to be completely invariant across trials of the experiment, then the model would make the undesirable prediction that every participant should have exactly the same response in a given trial type. To allow for population-level variation, the model responder makes a response by comparing the speaker probability against thresholds that are generated from sampling from Gaussian distributions. We thus allow for both population-level and individual level-variation, on the assumption that this sampling procedure takes place whenever a participant is asked to evaluate an utterance in the TVJT.[8]

In order to evaluate the RSA-based threshold model, we conducted a Bayesian data analysis. This allowed us to simultaneously generate model predictions and infer likely parameter values, by conditioning on the TVJT data from Jasbi et al. (separately for each of the four response-option conditions of the experiment) and integrating over the free parameters. Each model assumes uniform priors over utterances and world states as above. We infer the Gaussian threshold distribution parameters and alpha optimality parameters from uniform priors over parameter values using MCMC sampling (observing - for every sample of possible parameter values the expected proportion of responses in that trial type and comparing that distribution to the empirically-observed pattern of response).[9] Additionally, for the Quater-

[5]The effect of this is similar to the use of QUD projection functions for hyperbolic interpretations (Kao et al., 2014).

[6]Formalizing this production behavior is different from analyzing *why* imprecision exists (indeed, is pervasive) in linguistic communication. For the time being, we present this 'loose-talk' speaker model without a thorough assessment of its explanatory power.

[7]Because *cat or dog* is a possible S_1 production, and this choice lies in an entailment relation with the simple declarative guess *dog*, we also assign some probability to *dog* as a guess in this context - albeit lower probability than is assigned to the conjunctive guess *cat and dog*.

[8]We also introduce a random noise term in the parameter estimation such that the simulated responder makes random guesses on 1% of trials. This noise term is removed when running the model forward to make predictive estimations.

[9]We used WebPPL (Goodman and Stuhlmüller, 2014) for

Binary condition

α	σ	μ_{θ_1}
1.22	0.125	0.073

Ternary condition

α	σ	μ_{θ_1}	μ_{θ_2}
1.38	0.076	0.061	0.011

Quaternary condition

α	σ	μ_{θ_1}	μ_{θ_2}	μ_{θ_3}	PT
2.75	0.159	0.277	0.101	0.048	0.797

Quinary condition

α	σ	μ_{θ_1}	μ_{θ_2}	μ_{θ_3}	μ_{θ_4}	PT
4.38	0.099	0.184	0.042	0.005	0.002	0.437

Table 3: MAP estimates obtained from Bayesian data analysis, where α is the optimality parameter, σ and μ are Gaussian threshold distribution parameters, and PT is the probability with which the speaker samples from $P_{S_{PT}}$ rather than directly from P_{S_1}.

nary and Quinary conditions, we infer from a uniform prior the probability with which the speaker samples from $P_{S_{PT}}$ after sampling from P_{S_1}. The intuition for restricting the 'Partial Truth' manipulation to these conditions is that the behavioral patterns which this manipulation is intended to cover are only observed in these conditions.[10]

Posterior distributions over the parameter values are displayed in Figure 3, and model predictions using maximum a posteriori (MAP) estimates of the parameter values (Table 3) are plotted against Jasbi et al. (2019)'s results in Figure 1. Qualitatively, the model addresses each of the desiderata for an empirically adequate linking function discussed above. In all conditions, the model makes predictions for the full range of response options available to participants – thus addressing the issue of **intermediate judgments**. At the same time, the model addresses the issue of **population-level variation**: sampling threshold values from Gaussian distributions allows different judgments in the population for a given utterance (while keeping the speaker production probability of that utterance constant).

Recall that in the Quaternary and Quinary conditions, there was an asymmetry in the judgment of underinformative disjunctions versus underinformative simple declaratives. The model makes use of the 'Partial Truth' speaker function in order to adjust the underlying speaker production probabilities - and hence the distribution of predicted response options - for these utterances. The 'Partial Truth' function also boosts the production probability of strictly false conjunctions, allowing the model to predict responses other than 'Wrong' for this trial type. Thus, the 'Partial Truth' enrichment helps to address both **scalar diversity** and **endorsement of false utterances**.[11]

The correlation between empirical observations and model predictions is high (Adj. $R^2 > 0.9$ in all conditions), suggesting that the threshold responder model is a good model of TVJT behavior overall. Nevertheless, the model makes some undesirable predictions. For example, it over-predicts rates of 'Neither' responses in the Quinary condition. Empirically, this response tended to be disfavored relative to positive and negative response options, for example in the case of strictly false *cat and dog* guesses. The model assumes that the labeling of the response options should have no particular effect on selection, but future work should engage with this assumption.

4 Discussion and Conclusion

Based on a single underlying probabilistic model of pragmatic competence, the presented threshold responder model provides a level of empirical coverage for TVJT data unavailable to existing linking models rooted in the categorical view of scalar implicature. The contribution of this paper is twofold: methodologically, we present this analysis as a proof-of-concept approach to modeling TVJT data for researchers in experimental semantics/pragmatics. We see the presented behavioral model as a starting point for future quantitative analytic work in the TVJT domain – a model against which future models may be assessed.[12]

On the theoretical side, the cognitive model that forms the basis for the behavioral model is non-neutral in its assumptions. In particular, it assumes that TVJT behavior is the result of reasoning about probabilistic utterance choices that

MCMC inference, with 5000 samples (plus a lag of 10 iterations between samples) and a burn-in time of 20,000 iterations. We computed maximum a posteriori values from the marginal posterior distributions over parameter values using the `density` function in R.

[10] We speculate that there may be a link between increasing the number of response options and participants' increased expectation of Partial Truth speaker behavior, which may have been strengthened by the fact that the Quaternary and Quinary conditions explicitly made reference to gradient levels of correctness (i.e. 'Kinda Right' / 'Kinda Wrong'). But this speculation warrants future investigation.

[11] We leave further investigation of the 'Partial Truth' function - in particular its extension to an analysis of linguistic imprecision as sketched above - to future work.

[12] For example, one could in principle link the threshold model to pragmatic listener probabilities of meanings given utterances rather than to speaker production probabilities given intended meanings (as we do in this paper).

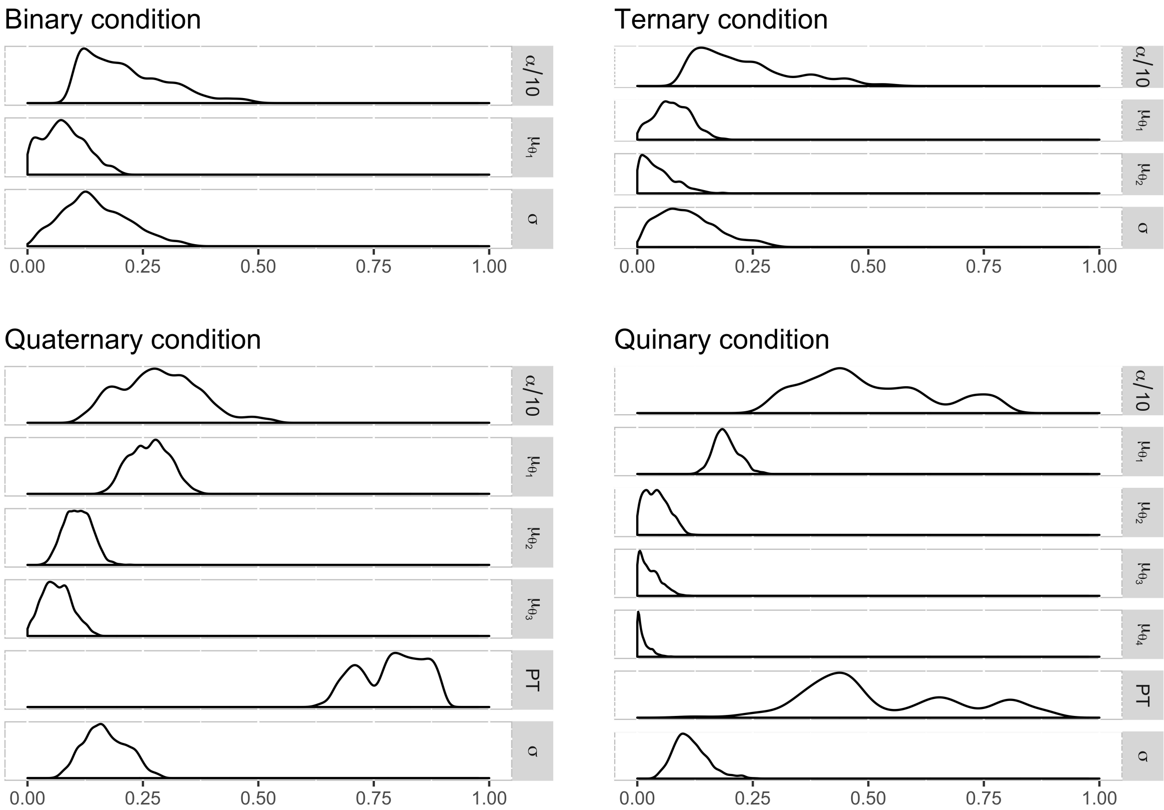


Figure 3: Normalized marginal posterior distributions over parameter values for the threshold responder model in each experimental condition. Note that the posterior distribution for the optimality parameter α has been rescaled for the purposes of this visualization.

are the result of trading off (contextual) utterance informativeness and cost. Under this view, not only does TVJT behavior not quantify implicature rates; the very notion of an implicature evaporates. Rather than finding this undesirable, we believe that this framework allows for more rigorous engagement with the complexities of linking theoretical constructs to behavior (see also Franke 2016), an area of some dearth in experimental semantics/pragmatics.

—

References

Lewis Bott and Ira Noveck. 2004. Some utterances are underinformative: The onset and time course of scalar inferences. *Journal of Memory and Language*, 51(3):437–457.

Emmanuel Chemla and Benjamin Spector. 2011. Experimental evidence for embedded scalar implicatures. *Journal of Semantics*, 28(3):359 – 400.

Wim De Neys and Walter Schaeken. 2007. When people are more logical under cognitive load - dual task impact on scalar implicature. *Experimental Psychology*, 54(2):128–133.

Judith Degen and Noah D Goodman. 2014. Lost your marbles? The puzzle of dependent measures in experimental pragmatics. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 397–402.

Judith Degen and Michael K. Tanenhaus. 2015. Processing scalar implicature A constraint-based approach. *Cognitive Science*, 39(4):667–710.

Ryan Doran, Gregory Ward, Meredith Larson, Yaron McNabb, and Rachel E. Baker. 2012. A novel experimental paradigm for distinguishing between what is said and what is implicated. *Language*, 88:124–154.

Michael C. Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336:998.

Michael Franke. 2016. Task types, link functions & probabilistic modeling in experimental pragmatics. In *Preproceedings of Trends in Experimental Pragmatics*.

Bart Geurts and Nausicaa Pouscoulous. 2009. Embedded implicatures?!? *Semantics and Pragmatics*, 2:1–34.

Noah D Goodman and Andreas Stuhlmüller. 2013. Knowledge and implicature: modeling language understanding as social cognition. *Topics in Cognitive Science*, 5(1):173–84.

Noah D Goodman and Andreas Stuhlmüller. 2014. The Design and Implementation of Probabilistic Programming Languages. http://dippl.org. Accessed: 2019-8-8.

Herbert Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:41–58.

Laurence Horn. 1972. *On the Semantic Properties of the Logical Operators in English*. Ph.D. thesis, UCLA.

Masoud Jasbi, Brandon Waldon, and Judith Degen. 2019. Linking hypothesis and number of response options modulate inferred scalar implicature rate. *Frontiers in Psychology*, 10:189.

Justine Kao, J. Wu, Leon Bergen, and Noah D Goodman. 2014. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences of the United States of America*, 111(33):12002–12007.

Napoleon Katsos and Dorothy V M Bishop. 2011. Pragmatic tolerance: implications for the acquisition of informativeness and implicature. *Cognition*, 120(1):67–81.

Peter Lasersohn. 1999. Pragmatic halos. *Language*, pages 522–551.

Ira Noveck. 2001. When children are more logical than adults: experimental investigations of scalar implicature. *Cognition*, 78(2):165–188.

Ira Noveck and Andres Posada. 2003. Characterizing the time course of an implicature: an evoked potentials study. *Brain and Language*, 85(2):203–210.

Anna Papafragou and Julien Musolino. 2003. Scalar implicatures: experiments at the semanticspragmatics interface. *Cognition*, 86:253–282.

Christopher Potts, Daniel Lassiter, Roger Levy, and Michael C Frank. 2015. Embedded implicatures as pragmatic inferences under compositional lexical uncertainty. *Journal of Semantics*, 33(1975):755–802.

Michael K. Tanenhaus. 2004. On-line sentence processing: past, present and future. In Manuel Carreiras and Charles Clifton, editors, *On-line sentence processing: ERPS, eye movements and beyond*, pages 371–392. Psychology Press, London, UK.

Bob van Tiel, Emiel van Miltenburg, Natalia Zevakhina, and Bart Geurts. 2014. Scalar diversity. *Journal of Semantics*.

What Don't RNN Language Models Learn About Filler-Gap Dependencies?

Rui P. Chaves
Linguistics Department
University at Buffalo – SUNY
rchaves@buffalo.edu

Abstract

In a series of experiments Wilcox et al. (2018, 2019b) provide evidence suggesting that general-purpose state-of-the-art LSTM RNN language models have not only learned English filler-gap dependencies, but also some of their associated 'island' constraints (Ross, 1967)). In the present paper, I cast doubt on such claims, and argue that upon closer inspection filler-gap dependencies are learned only very imperfectly, including their associated island constraints. I conjecture that the LSTM RNN models in question have more likely learned some surface statistical regularities in the dataset rather than higher-level abstract generalizations about the linguistic mechanisms underlying filler-gap constructions.

1 Introduction

Recurrent Neural Networks (RNNs) are a class of abstract neural network where the connections between nodes consist of a directed graph along a temporal sequence. This architecture allows node outputs at current time step to depend on the current input as well as on the previous output state. Thus, the network can exhibit temporal dynamic behavior, since the internal state of the system is a kind of memory that can be used to process subsequent input. Such models are therefore well-suited for natural language tasks, among others. RNNs with a Long Short-Term Memory (LSTM) architecture have a far more elaborate and selective form of memory. A common LSTM node is composed of a cell, an input gate, an output gate and a forget gate. Such gates enable RNN nodes to remember values over arbitrary time intervals and the three gates regulate the flow of information into and out of the nodes.

LSTM RNNs are therefore better suited than plain RNNs to model long-distance dependencies of the kind found in natural languages (Linzen et al., 2016; Gulordava et al., 2018; Bernardy and Lappin, 2017). This includes filler-gap dependencies like (1), where the *wh*-phrase *what* is interpreted as the object of *do*, even though the two words are separated by four clausal boundaries as indicated by square brackets.

(1) What$_i$ do you think [the students will say [they believe [the TA claimed [he was trying to do $\_{i}$]]]]?

I refer to the 'extracted' phrase as the *filler* and to the canonical position where it would otherwise be realized as the *gap*, signaled via an underscore. The filler-gap dependency is the semantic and syntactic linkage that must be established between the filler and its *in situ* canonical location in order for such utterances to be interpretable.

1.1 Learning Filler-Gap dependencies

Recently, Chowdhury and Zamparelli (2018) provide some evidence that LSTM RNNs can store information about the filler phrase, and detect that the probability of the sentence-final NP in examples like (2) is low because of the presence of a filler-gap dependency.

(2) Who$_i$ should Mia discuss $\_{i}$ / *this candidate.

Wilcox et al. (2018) improve on this work, and propose a Surprisal-based (Hale, 2001; Levy, 2008) differences-within-differences design to measure the ability of the RNN to learn filler-gap dependencies, using a factorial design as in (3).

(3) a. I know that the lion devoured a gazelle **at** sunrise.
[NO WH-LICENSOR, NO GAP]

b.*I know what the lion devoured a gazelle **at** sunrise.
[WH-LICENSOR, NO GAP]

c.*I know that the lion devoured _ **at** sunrise.
[NO WH-LICENSOR, GAP]

d. I know what$_i$ the lion devoured _$_i$ **at** sunrise.
[WH-LICENSOR, GAP]

Wilcox et al. define $S(w)$ as the surprisal of a given word w, estimated in terms of the log inverse probability of w according to the RNN's hidden state softmax activation h before consuming w, given all previous words in the sentence:

(4) $S(w) = -log_2\ p(w|h)$

If the model has learned to represent filler-gap dependencies, then the surprisal of the proposition *at* in (3a) should be a small number, since the probability of *at* in this context is high, and the surprisal of 'at' in (3b) should be a large number, since the probability of 'at' in this context is low. Consequently, their difference $S(3b) - S(3a)$ should yield a large positive number. Similarly, $S(3d) - S(3c)$ should yield a large negative number, and the full **licensing interaction** $(S(3b) - S(3a)) - (S(3d) - S(3c))$ should be a large positive number. This licensing interaction represents how well the network learns both parts of the licensing relationship: a positive wh-licensing interaction means the model represents a filler-gap dependency between the wh-word and the gap site; a licensing interaction indistinguishable from zero indicates no such dependency. Wilcox et al. find that typical models show about 4 bits of licensing interaction in simple examples like (3).

Using this design, Wilcox et al. (2019b) found that LSTM RNNs can maintain filler-gap dependencies across up to four clausal boundaries, not unlike the ones in (1). Two models were used for these experiments: (i) the model in Gulordava et al. (2018) – henceforth the **Gulordava model** – which was trained on 90 million tokens of English Wikipedia, and has two hidden layers of 650 units each; and (ii) Jozefowicz et al. (2016) – henceforth the **Google model** – which was trained on the One Billion Word Benchmark (Chelba et al., 2013), has two hidden layers with 8196 units each, and employs a character-level convolutional neural network.

But more recently Da Costa and Chaves (2020) shows that the Gulordava and Google LSTM models have learned filler-gap dependencies only very imperfectly. In particular, the models completely

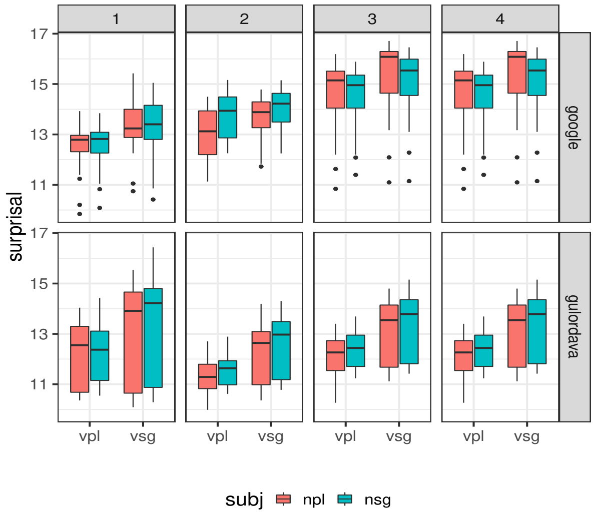


Figure 1: Surprisal at the gap-agreeing verb in 'which' interrogatives across embedding levels (LSTM RNNs)

failed to learn that filler-gap constructions also impose agreement dependencies like those in (5). In such constructions, the singular/plural number information of the extracted phrase must match that of the verb from which the extraction takes place.

(5) a. They wondered which lawyer I think you said _ was/*were upset.

b. They wondered which lawyers I think you said _ *was/were upset.

Following the same factorial approach and code of Wilcox et al. (2018), Da Costa and Chaves (2020) extracted the softmax activation of the verbs *were*/*was* in 20 items like those illustrated in (6), up to four levels of clausal embedding.

(6) a. Someone wondered which lawyer(s) I think was/were ...
[$N_{sg/pl}$, LEVEL1, $V_{sg/pl}$]

b. Someone wondered which lawyer(s) I think you said was/were ...
[$N_{sg/pl}$, LEVEL2, $V_{sg/pl}$]

c. Someone wondered which lawyer(s) I think you said you thought was/were ...
[$N_{sg/pl}$, LEVEL3, $V_{sg/pl}$]

d. Someone wondered which lawyer(s) who people believe I think you said you thought was/were ...
[$N_{sg/pl}$, LEVEL4, $V_{sg/pl}$]

The results in Figure 1 show that both the Gulordava and the Google models failed. Had the

LSTM RNNs succeeded at this task, the conditions where the noun and verb agree (i.e. N_{pl}-V_{pl} and N_{sg}-V_{sg}) would be lower in surprisal than the conditions where the agreement is mismatched (i.e. N_{pl}-V_{sg} and N_{sg}-V_{pl}). Note also that in the Google model surprisal increased with the level of embedding, so that the correct verb form is more unexpected in level 4 than the incorrect verb forms in levels 1 and 2. Da Costa and Chaves (2020) tested other types of construction and the results are equally bad, suggesting that the Gulordava and Google models have not learned the morphosyntax of filler-gap dependencies, even though they were trained on datasets larger than what a child learner is exposed to; according to Atkinson et al. (2018), children begin to exhibit adult-like active formation of filler-gap dependencies by age 6.

1.2 Learning Island Constraints

Wilcox et al. (2018, 2019b) in addition claim that the Gulordava and Google models have learned certain constraints on filler-gap dependencies known as **Islands** (Ross, 1967). In particular, Wilcox et al. claim that the models learn that the subordinate clauses introduced by *whether* have reduced acceptability as in (7a), that relative clauses and adverbial adjuncts are difficult to extract from as in (7b,c), and that conjuncts and the left branches of NP are not possible to extract, as in (7d,e). All reported examples below are from Wilcox et. al experiments. Square brackets indicate the island-establishing environments.

(7) a.* I know what Alex said [whether your friend devoured _ at the party].
(**Wh-Island**)

b.*I know (that/what/who) the family bought the painting [that depicted _ last year].
(**Complex NP Constraint Island**)

c.*I know what the patron got mad [after the librarian placed _ on the wrong shelf].
(**Adjunct Constraint Island**)

d.*I know what the man bought [the painting and _] at the antique shop.
(**Conjunct Constraint island**)

e.*I know what color you bought [_ car] last week.
(**Left Branch Constraint island**)

However, Wilcox et.'s claims are too strong. First, most of these island constraints are more complex than Wilcox et. al's discussion suggest, and before it cannot be claimed that a model learns island constraints before all the associated conditions are shown to have been learned as well. For example, the Conjunct Constraint is but a piece of a larger set of constraints that are specific to coordination, known as the Coordinate Structure Constraint (CSC). The CSC consists of the Conjunct Constraint, the Element Constraint, the ATB Exception, and the Asymmetric Exception; see Kehler (2002, Ch.5) for a detailed overview and an account of most of these constraints that is based on pragmatic discourse relations.

The Complex NP Constraint (CNPC) is similarly complex. First, it is not restricted to relative clauses: nouns that semantically introduce propositional complements like in *the claim that Robin stole a book* also induce such extraction limitations (e.g. *$What_i$ did you reject the claim [that Robin stole $\__i$]?'). Second, it is also known that the CNPC vanishes in presentational relatives (i.e. in relatives that express assertions rather than pressupposed content), as we discuss below.

Moreover, some of the island constraints that Wilcox *et al.* probed are know to be weakened when the island phrase is untensed, and vanish altogether if there is a secondary (i.e. 'parasitic') gap outside the adjunct (Engdahl, 1983); see Phillips (2006) for experimental evidence. In sum, there is a complex array of facts that still need to be tested.

Finally, the Left Branch Constraint (LBC) items that Wilcox *et al.* used, like (7e), have a critical confound. The sentences are not licit even without the extraction (i.e. **what color car*). And since the sentences are ill-formed, with or without extraction, it remains unclear whether the RNNs have or not learned the LBC.

But even conceding that the results are overall on the right track, there is one final problem. Both the Gulordava and Google models failed to learn that extraction from subject phrases (phrasal or clausal) is hampered, as illustrated in (8).

(8) a.*I know who [the painting by _] fetched a high price at auction.
(**Subject Constraint Island**)

b.*I know who [for the seniors to defeat _] will be trivial.
(**Sentential Subject Constraint Island**)

The difficulty in learning clausal Subject Island effects is unexpected because such islands are much stronger than Wh-islands. Not only the oddness induced by a Wh-island constraint violation is less pronounced than that of clausal Subject islands, but also because counterexamples to the former are much easier to find. Compare (7) with the acceptable counterpart in (9).

(9) Which shoes are you wondering [whether you should buy _]?

See Abrusán (2014, Ch.4) for strong evidence that Wh-islands and their exceptions are contingent on subtle semantic-pragmatic factors, not syntax. Indeed, there is growing evidence that many island constraints are at least in part due to non-syntactic factors, including pragmatics and processing biases; see Chaves and Putnam (2020) for a detailed overview. For example, counterexamples have been noted in the literature to all of the island constraints probed by Wilcox et al., with the exception of the Conjunct Constraint and the Left Branch Constraint islands; see Hofmeister and Sag (2010) and references cited. This includes Subject Islands involving VP subjects, as in the attested data in (10). See Huddleston et al. (2002, 1093,1094), Santorini (2007), and Chaves (2013) for more attestations.

(10) a. In his bedroom, which [to describe _ as small] would be a gross understatement, he has an audio studio setup.
[pipl.com/directory/name/Frohwein/Kym]

b. They amounted to near twenty thousand pounds, which [to pay _] would have ruined me. (Benjamin Franklin, William Temple Franklin and William Duane. 1834. Memoirs of Benjamin Franklin, vol 1. p.58)
[archive.org/details/membenfrank01frankrich]

c. The (...) brand has just released their S/S 2009 collection, which [to describe _ as noticeable] would be a sore understatement.
[missomnimedia.com/2009/page/2/?s=art+radar&x=0&y=0]

d. Because this does purport to be a food blog, I will move from the tv topic to the food court itself, which [to describe _ as impressive] would be an understatement.
[phillyfoodanddrink.blogspot.com/2008/06/foodies-food-court.html]

All of these counterexamples involve restrictive relative clauses, suggesting that the Subject Condition is sensitive to pragmatics (Abeillé et al., 2018; Chaves and Dery, 2019).

The point here is a cautionary one: many island constraints are not absolute, and come with a complex array of patterns, many of which are still poorly understood. It cannot be claimed that a given language model has learned an island constraint before showing that both the negative and the positive cases (if any exist) have been correctly learned as well.

Note also that the Gulordava and the Google models did not perform in the same way at learning these island constraints: whereas the Google model failed to learn CNPC islands when the word 'that' appears instead of 'who/what', the Gulordava model failed to learn Wh-Islands. The performance of the Google was not significantly better that Gulordava's even though the former was originally trained with ten times more data than the latter, contained ten times as many hidden units, and used character CNN embeddings. This again suggests that something fundamental about filler-gap dependencies is being missed.

The question then becomes: are these models actually learning filler-gap dependencies or are they simply learning surface-based contingencies that have little to do with the underlying syntactic and semantic mechanisms that cause island phenomena? As Jo and Bengio (2017) demonstrate, neural networks tend to learn surface statistical regularities in the dataset rather than higher-level abstract concepts; for adversarial research showing this to be the case in the language domain see Jia and Liang (2017) and Iyyer et al. (2018), for instance. Indeed, Marvin and Linzen (2018) found that LSTM RNNs fail to learn reflexive pronoun agreement and negative polarity licensing, and Wilcox et al. (2019a) showed that such models learn center-embedding dependencies only imperfectly. In the remainder of this paper the same models, code and licensing interaction approach of Wilcox et al. (2018) is used to provide evidence suggesting that these LSTM RNNs merely capture partial and superficial morphosyntactic properties of filler-gap dependency constraints. The present results are consistent with those of Wilcox et al. (2019a), in which these models are not fully able to suppress expectations for gaps inside at least some island environments and recover them later.

2 Extraction from Relative Clauses

Wilcox et al. (2018) found that evidence suggesting that both the Google and the Gulordava models have learned the CNPC. However, the CNPC is not without principled exceptions. It is well-known that CNPC effects systematically vanish in existential relative clauses (Erteschik-Shir and Lappin, 1979; McCawley, 1981; Chung and McCloskey, 1983) as in (11). See Kush et al. (2013) for experimental evidence that existential relatives are not island inducing syntactic environments.

(11) a. This is the kind of weather that there are [many people who like _]. (Erteschik-Shir and Lappin, 1979)

b. There were several old rock songs that she and I were [the only two who knew _]. (Chung and McCloskey, 1983)

c. John is the sort of guy that I don't know [a lot of people who think well of _]. (Culicover, 1999, 230)

d. Which diamond ring did you say there was [nobody in the world who could buy _]? (Pollard and Sag, 1994, 206)

Such relatives are special in that they express assertions rather than presupposed content, and the extraction is thus arguably acceptable because the referent that is questioned is part of the content that is asserted and at-issue (Goldberg, 2013). It should be relatively easy for the models to use the *there be* sequence as a cue that these constructions are different from other relatives. If Google and Gulordova's RNN models have learned the CNPC rather than superficial contingencies then the existence of a second gap inside an existential relative should not cause a large spike in surprisal and the licensing interaction should be small, or ideally, close to zero. For this purpose 18 experimental items were taken from Kush et al. (2013) and adapted to the present task, using the methodology as Wilcox et al. A sample is in (12).[1]

(12) a. It was known that there were many mathematicians who worked on the project **for** years.
[NO WH-LICENSOR, NO GAP]

[1] Only verbs that strongly require complements were employed, and that-relatives were avoided given that the models have difficulty with them according to Wilcox et al. (2018).

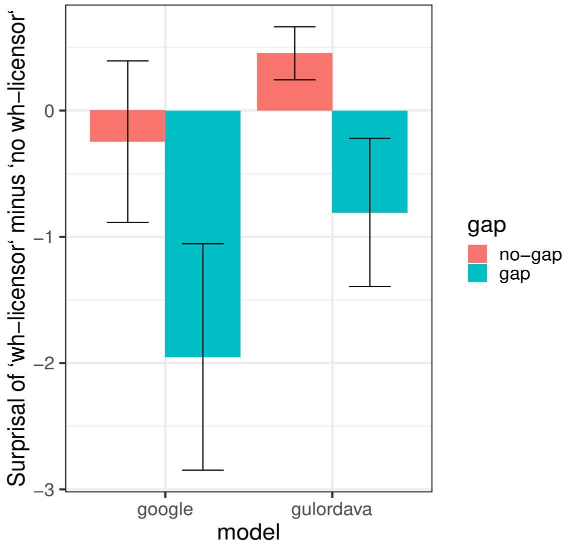


Figure 2: Licensing Interaction in Existential Relatives

b.*This was the problem which there were many mathematicians who worked on the project **for** years.
[WH-LICENSOR, NO GAP]

c.*It was known that there were many mathematicians who worked on _ **for** years.
[NO WH-LICENSOR, GAP]

d. This was the problem which there were many mathematicians who worked on _ **for** years.
1 [WH-LICENSOR, GAP]

Ideally, the no-gap condition interaction $S(12\text{b}) - S(12\text{a})$ should be a positive number, and the gap condition interaction $S(12\text{d}) - S(12\text{c})$ a negative number. As the graphs in Figure 2 indicate, this is what was found for the Gulordava model, but not for Google's. In the latter, the no-gap condition is indistinguishable from zero (t = -0.75, p = 0.46) suggesting that the latter model overlooks the subject gap. That said, the full wh-licensing interaction values are clearly positive, and in the order of about 1.5 bits. This is much lower than the 4 bits found by Wilcox et al. (2018), but nonetheless suggests that at least some aspects of the filler-gap dependency are detected by the models. Many other attempts were made to arrive at stronger results, with different materials, but the results invariably had similar outcomes, with the 'no-gap' bars either being indistinguishable from zero or negative. I now move on to islands which are not as strongly correlated with surface cues.

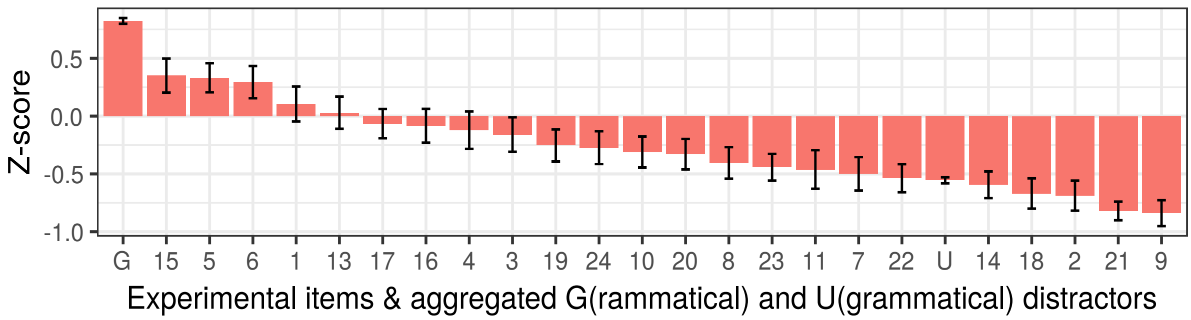


Figure 3: Acceptability ratings by item (with grammatical (G) and ungrammatical (U) distractors aggregated)

3 Extraction from Adjunct Clauses

Wilcox et al. (2018) probed the strongest type of adjunct island (tensed adjuncts), traditionally regarded as exceptionless since Huang (1982). But recent work has revealed that exceptions do exist; see Kluender (1998, 267), Truswell (2011, 175, ft.1), Levine and Hukari (2006, 287), and Goldberg (2006, 144). For example, Sprouse et al. (2016) found no evidence of an island effect in examples like (13), in terms of sentence acceptability rating, but found strong evidence of island effects in other adjunct island examples.

(13) I called the client [who]$_i$ the secretary worries [if the lawyer insults _$_i$].
(Sprouse et al., 2016)

Similarly, Müller (2017) experimentally shows that Swedish conditional adjuncts seem to yield much weaker island effects than causal adjuncts, and Kohrt et al. (2018) found experimental evidence that (non-clausal) English adjunct islands are contingent on semantic factors. In more recent work, Chaves and Putnam (2020) provide experimental evidence suggesting that Mueller's results likely extend to English as well. Chaves and Putnam (2020) report a sentence acceptability experiment with 24 items falling into three conditions, illustrated in (14).

(14) a. Who$_i$ did Sue blush [when she saw _$_i$]? [TEMPORAL ADJUNCT]

b. What$_i$ did Tom get mad [because Phil forgot to say _$_i$]? [CAUSAL ADJUNCT]

c. What$_i$ does Evan get grumpy [if he is told to do _$_i$]? [CONDITIONAL ADJUNCT]

I what follows I briefly describe this experiment in more detail, with the aim of repurposing the items for a counterpart experiment using the Gulordava and Google models. Each item was interspersed and pseudo-randomized with 36 filler phrases, half of which are ungrammatical, as illustrated in (15). The grammatical distractors were immediately followed by Yes/No comprehension questions, and the mean comprehension question accuracy was 86%.

(15) a.*Who does the union identify as having most recently fired from _?

b. What did the editor recommend should be revised _?

Chaves and Putnam analyzed data from 38 English native speakers, who were asked to rate the acceptability of each experimental item on a 5-point Likert scale. There was a wide range of acceptability scores, from fairly high in the acceptability scale to very low, as seen in Figure 3. The (aggregate) ratings for the grammatical (G) and the ungrammatical (U) distractors are included, for comparison. Conditional adjuncts were clustered at the high end of the ratings, temporal adjuncts in the middle, and causal adjuncts at the bottom.

I now describe how the stimuli from this experiment was repurposed to the same task that Wilcox et al. (2018) employed. The top 5 human-rate rated items (High Acceptability condition) received a mean acceptability of 3.30 ($SD = 0.2$), and the bottom human-rated 5 rated items (Low Acceptability condition) received a mean acceptability of 1.95 ($SD = 0.13$). These 10 items were selected and adapted to the $3 \times 2 \times 2$ factorial licensing interaction methodology of Wilcox et al. (2018). The counterparts of the item in (14c) are shown in (16) and (17) for illustration. In a nutshell, all items were embedded under 'I know' and

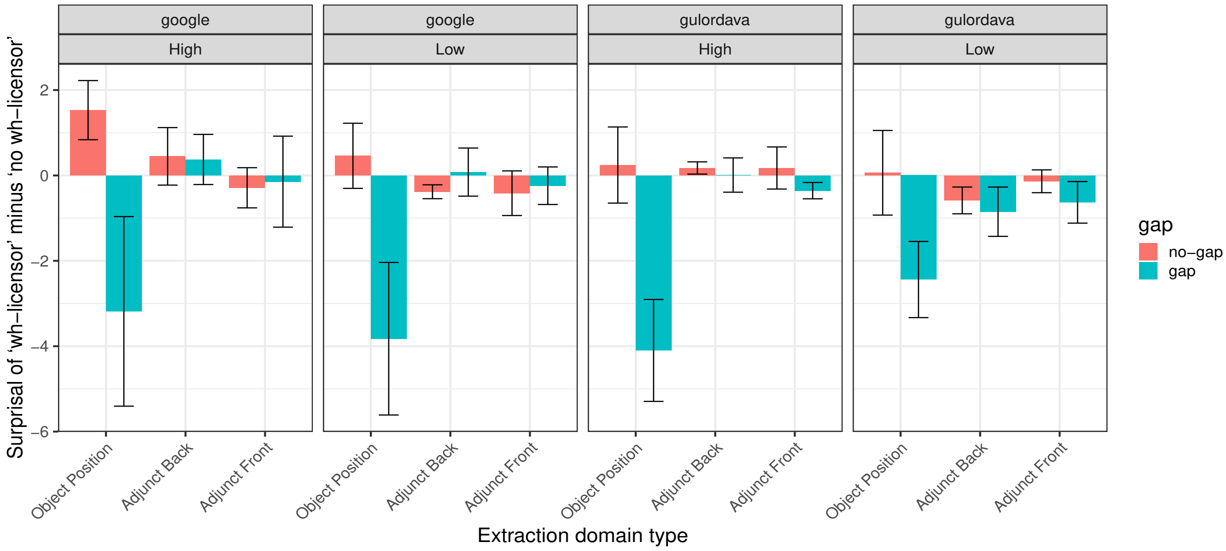


Figure 4: Effect of extraction site on wh-licensing interaction for adjunct islands, across high/low acceptability

all proper names were replaced with pronouns. In the Object condition there is no adjunct clause.

(16) a. I know that they usually are told to do the homework **in** the morning.
[OBJECT, NO WH-LICENSOR, NO-GAP]

b.*I know what they usually are told to do the homework **in** the morning.
[OBJECT, WH-LICENSOR, NO-GAP]

c.*I know that they usually are told to do _ **in** the morning.
[OBJECT, NO WH-LICENSOR, GAP]

d. I know what they usually are told to do _ **in** the morning.
[OBJECT, WH-LICENSOR, GAP]

In the Adjunct back condition there is an adjunct clause at the end of the sentence, as in (17). Following Wilcox et al. (2018), there was a third condition where the adverbial clause is fronted, and appears immediately after the complementizer *that* rather than at the end of the utterance.

(17) a. I know that the kids get grumpy if they are told to do the homework **in** the morning.
[ADJUNCT BACK, NO WH-LICENSOR, NO-GAP]

b.*I know what the kids get grumpy if they are told to do the homework **in** the morning.
[ADJUNCT BACK, WH-LICENSOR, NO-GAP]

c.*I know that the kids get grumpy if they are told to do _ **in** the morning.
[ADJUNCT BACK, NO WH-LICENSOR, GAP]

d. I know what the kids get grumpy if they are told to do _ **in** the morning.
[ADJUNCT BACK, WH-LICENSOR, GAP]

If the Gulordava and Google models have learned the subtleties of the tensed Adjunct Constraint then the filler-gap dependencies in the High Acceptability condition items should have a significantly lower surprisal than the Low Acceptability condition items. In order to access this, the surprisal of the word after the critical region was measured. Focusing on the object items first, interactions of the type $S(16\text{b}) - S(16\text{a})$ should ideally result in a positive number, however, for both High acceptability or Low acceptability items. This was the case in the Google model, but not for the Gulordava model, as Figure 4 shows; perhaps the latter model discovered that a gap after the preposition in (16b) is not necessarily out of the question. $S(16\text{d}) - S(16\text{c})$ yielded the expected highly negative values, as illustrated by the long teal bars.

Moving on to the Adjunct back items, the interactions of the type $S(17\text{b}) - S(17\text{a})$ should ideally result in a positive number as usual, contrary to fact, and $S(17\text{d}) - S(17\text{c})$ should ideally result in a negative number in the High acceptability condition and cancel out in the Low acceptability conditions. Neither result occurred because the interaction values were centered around zero. The full licensing interaction $(S(17\text{b}) - S(17\text{a})) - (S(17\text{d}) - S(17\text{c}))$ is shown in Figure 5. None of the Adjunct front/back High/Low conditions is statistically distinguishable from zero, although significance is approached ($t = 2.73, p = 0.052$) in the

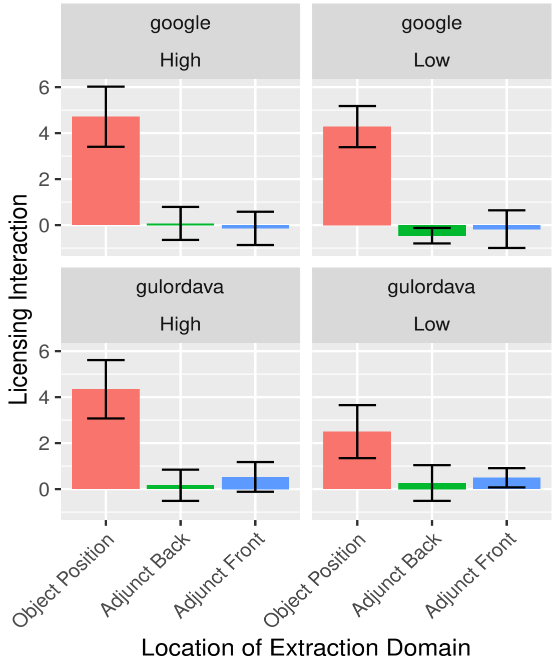


Figure 5: Full licensing interaction for Adjunct Islands

case of Adjunct front High for Gulordava.

In sum, all extractions from clausal adjuncts are ultimately deemed islands environments by the models, contrary to the human judgments.

4 Extraction from Negative Phrases

Negative Islands are perhaps the clearest type of island in which semantic and pragmatic factors play a key role. Consider the examples in (18).

(18) a.*Which country weren't you born in _?

b.*How many kids don't you have _?

c.*How fast didn't John drive _?

The question in (18a) presupposes that the addressee was born in all countries but one, which is contrary to world knowledge, and therefore infelicitous (Kuno and Takami, 1997). Hence, the oddness vanishes if the verb is not a one-time predicate, as in (19).

(19) Which country haven't you visited _ yet?

The oddness of the degree questions in (18b,c) is due to an analogous reason; see Abrusán (2011) for detailed discussion. It is again clear that the oddness is caused by semantic factors, since the introduction of existential modals makes the island effect vanish (Fox and Hackl, 2006):

(20) a. How many kids can't you have _?

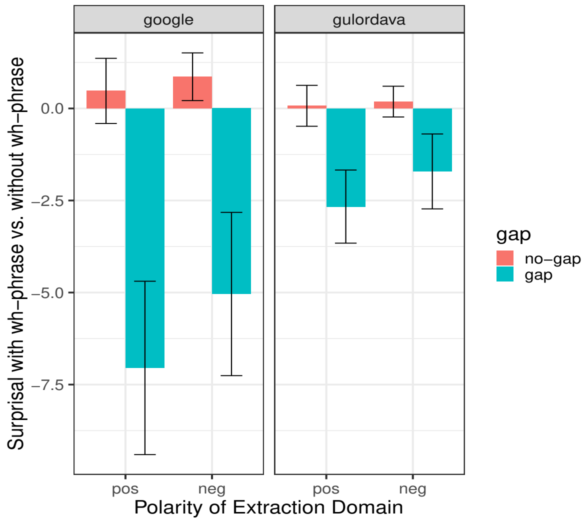


Figure 6: Wh-licensing in negative phrases

b. How fast is John required not to drive _?

In order to evaluate whether RNNs are sensitive to such effects 14 items were constructed in a $2{\times}2{\times}2$ design, as illustrated in (21). The verb is negated in items in the negative (NEG) condition.

(21) a. I wonder if the owner of the truck has (not) driven at this speed **during** the race. [NO WH-LICENSOR, POS/NEG, NO GAP]

b.*I wonder how fast the owner of the truck has (not) driven at this speed **during** the race. [WH-LICENSOR, POS/NEG, NO GAP]

c.*I wonder if the owner of the truck has (not) driven at _ **during** the race. [NO WH-LICENSOR, POS/NEG, GAP]

d. I wonder how fast the owner of the truck has (*not) driven at _ **during** the race. [WH-LICENSOR, POS/NEG, GAP]

The results are shown in Figure 6. The interaction S(21b) $-$ S(21a) should have resulted in a moderate-to-large positive numbers, regardless of the presence of negation. In other words, the red bars should be positive and not overlap with zero. This was not true of either model, especially for Gulordava. Conversely, S(21d)$-S$(21c) should have yielded a moderate-to-large negative number in the pos(itive) condition but obtain a significantly higher value in the neg(ative) condition (ideally, close to zero). However, there was no statistically significant difference between the interaction values across the two island conditions (pos and neg) for the Google model ($t = 0.3$, $p = 0.73$)

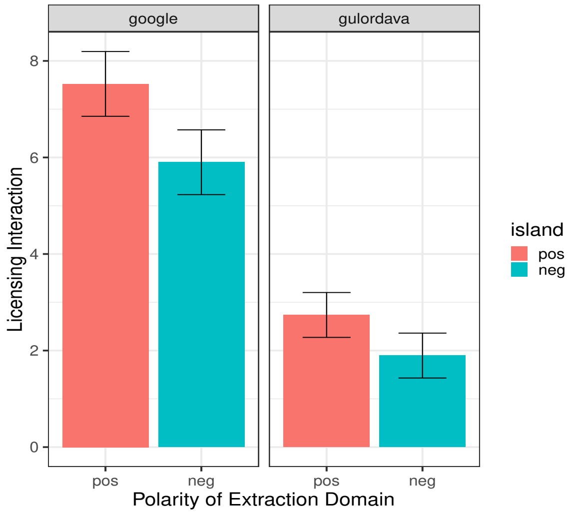


Figure 7: Full licensing interaction for negative islands

nor for the Gulordava model ($t = 1.11, p = 0.27$). The full interactions are shown in Figure 7. Had Negative Islands been learned, the teal bars would be centered around zero, like those in in Figure 5.

5 Discussion

The claim that sate-of-the-art LSTM RNNs models have learned filler-gap dependencies and islands is premature on both linguistic and experimental grounds. First, the linguistic constraints in question are far more complex than what extant studies consider. Second, there is evidence that these models only learn partial contingencies about filler-gap dependencies, which suggests that the actual linguistic mechanism that underlies such long-distance phenomena is not accessible to the model.

The problem is arguably not due to a lack of data. The training datasets for Gulordava and Google are unrealistically large when compared to the amount of linguistic input the average child is exposed to (Atkinson et al., 2018). Similarly, the problem is not likely to be due to lack of expressivity, since this kind of model is Turing-complete; see Siegelmann and Sontag (1995) and Siegelmann (1999, 29–58) for proofs and examples, as well as Hornik et al. (1989) and Lu et al. (2017) for detailed discussion about Cybenko's universal approximation theorem.

The present findings suggest that model size and training regimen yield diminishing returns, and that there is a more fundamental factor preventing such systems to learn filler-gap dependencies. The problem likely stems from the fact that filler-gap dependencies are not merely surface string patterns: they involve rich morphological, syntactic and semantic dependencies which crucially interact with pragmatics and world knowledge, thus far absent from training. Most crucially, many island phenomena seems to be sensitive to semantic and pragmatic constraints, including the Subject Constraint (Chaves and Dery, 2019; Abeillé et al., 2018), the Adjunct Constraint (Truswell, 2011; Müller, 2017; Kohrt et al., 2018; Goldberg, 2013), the Complex NP Constraint (Erteschik-Shir and Lappin, 1979; Goldberg, 2013), the Coordinate Structure Constraint (Kehler, 2002, Ch.5), Wh-Islands Abrusán (2014, Ch.4), Negative Islands (Abrusán, 2011), among others. See Chaves and Putnam (2020) for extensive discussion of these and other island effects.

In sum, it not clear how current neural models can learn island constraints from stringsets alone, precisely because of the subtle semantic and pragmatic properies that underpin the phenomena in question. The present findings are consistent with the fact that Marvin and Linzen (2018) found that LSTM RNNs fail to learn other complex phenomena such as reflexive pronoun agreement, negative polarity licensing, and center-embedding dependencies (Wilcox et al., 2019a).

All experimental items and statistical analysis scripts are made available online at https://github.com/RuiPChaves/LSTM-RNN-unbounded-dependency-experiments. The code to run the models is the same as Wilcox et al. (2018).

References

Anne Abeillé, Barbara Hemforth, Elodie Winckel, and Edward Gibson. 2018. A construction-conflict explanation of the subject-island constraint. 31th Annual CUNY Conference on Human Sentence Processing.

Márta Abrusán. 2011. Presuppositional and negative islands: A semantic account. *Natural Language Semantics*, 19:257–321.

Márta Abrusán. 2014. *Weak island semantics*. Oxford University Press, Oxford.

Emily Atkinson, Matthew W. Wagers, Jeffrey Lidz, Colin Phillips, and Akira Omaki. 2018. Developing incrementality in filler-gap dependency processing. *Cognition*, 179:132 – 149.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15(2):1–15.

Rui P. Chaves. 2013. An expectation-based account of subject islands and parasitism. *Journal of Linguistics*, 2(49):285–327.

Rui P. Chaves and Jeruen E. Dery. 2019. Frequency effects in subject islands. *Journal of Linguistics*, page 147.

Rui P. Chaves and Michael T. Putnam. 2020. *Unbounded Dependency Constructions: theoretical and experimental perspectives*. Oxford University Press, Oxford.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144.

Sandra Chung and James McCloskey. 1983. On the interpretation of certain island facts in GPSG. *Linguistic Inquiry*, 14:703–714.

Peter W. Culicover. 1999. *Syntactic Nuts: Hard Cases in Syntax*. Volume One of Foundations of Syntax. Oxford: Oxford University Press.

Jillian K. Da Costa and Rui P. Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *SCil*, page 10.

Alex Drummond. 2013. Ibex 0.3.7 manual. spellout.net/latest_ibex_manual.pdf.

Elisabet Engdahl. 1983. Parasitic gaps. *Linguistics and Philosophy*, 6:3–34.

Nomi Erteschik-Shir and Shalom Lappin. 1979. Dominance and the functional explanation of island phenomena. *Theoretical Linguistics*, 6:41–86.

Danny Fox and Martin Hackl. 2006. The universal density of measurement. *Linguistics and Philosophy*, 29:537–586.

Adele E. Goldberg. 2006. *Constructions at Work: the nature of generalization in Language*. Oxford: Oxford University Press.

Adele E. Goldberg. 2013. Backgrounded constituents cannot be extracted. In Jon Sprouse and Norbert Hornstein, editors, *Experimental Syntax and Island Effects*, pages 221–238. Cambridge: Cambridge University Press.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL*, pages 1195–1205.

John T. Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of NAACL-2001, Pittsburg, PA*, pages 159–166. ACL.

Philip Hofmeister and Ivan A. Sag. 2010. Cognitive constraints and island effects. *Language*, 86(2):366–415.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Cheng-Teh James Huang. 1982. *Logical relations in Chinese and the theory of grammar*. Ph.d. thesis, MIT.

Rodney D. Huddleston, Geoffrey Pullum, and Peter Peterson. 2002. *The Cambridge Grammar of the English Language*, chapter 12: Relative clause constructions and unbounded dependencies. Cambridge: Cambridge University Press.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL-HLT 2018*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Jo and Yoshua Bengio. 2017. Measuring the tendency of CNNs to learn surface statistical regularities. *CoRR*, abs/1711.11561.

Rafal Jozefowicz, Vinyals Oriol, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR*.

Andrew Kehler. 2002. *Coherence, Reference, and the Theory of Grammar*. Stanford, CA: CSLI Publications.

Robert Kluender. 1998. On the distinction between strong islands and weak islands: a processing perspective. In Peter W. Culicover and Louise McNally, editors, *Syntax and Semantics 29: The Limits of Syntax*, pages 241–279. New York, NY: Academic Press.

Annika Kohrt, Trey Sorensen, and Dustin A. Chacón. 2018. The real-time status of semantic exceptions to the adjunct island constraint. In *Proceedings of WECOL 2018: Western Conference on Linguistics*.

Susumu Kuno and Ken-ichi Takami. 1997. Remarks on negative islands. *Linguistic Inquiry*, 28:553–576.

David Kush, Akira Omaki, and Norbert Hornstein. 2013. Microvariation in islands? In Jon Sprouse and Norbert Hornstein, editors, *Experimental Syntax and Island Effects*, pages 239–264. Cambridge: Cambridge University Press.

Robert D. Levine and Thomas E. Hukari. 2006. *The unity of unbounded dependency constructions*. Stanford, CA: CSLI Publications.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 3(106):1126–1177.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqian Hu, and Liwei Wang. 2017. The expressive power of neural networks: A view from the width. In *Neural Information Processing Systems*, pages 6231–6239.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

James D. McCawley. 1981. The syntax and semantics of english relative clauses. *Lingua*, 53:99–149.

Christiane Müller. 2017. Extraction from adjunct islands in Swedish. *Norsk Lingvistisk Tidsskrift*, 35(1):6785.

Colin Phillips. 2006. The real-time status of island phenomena. *Language*, 82:795–823.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. Chicago: University of Chicago Press and Stanford: CSLI.

John R. Ross. 1967. *Constraints on Variables in Syntax*. Ph.d. dissertation, MIT, Cambridge, Massachusetts. [Published in 1986 as *Infinite Syntax!* Norwood, NJ: Ablex Publishing].

Beatrice Santorini. 2007. (Un?)expected movement. University of Pennsylvania. http://www.ling.upenn.edu/ beatrice/ examples/movement.html. Accessed: Jun 14 2019.

Hava T. Siegelmann. 1999. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Progress in Theoretical Computer Science. Birkhäuser, Boston, MA.

Hava T. Siegelmann and E. Sontag. 1995. On the computational power of neural nets. *Journal of Computer and Systems Sciences*, 50(1):132–150.

Jon Sprouse, Ivano Caponigro, Ciro Greco, and Carlo Cecchetto. 2016. Experimental syntax and the variation of island effects in english and italian. *Natural Language & Linguistic Theory*, 34(1):307–344.

Robert Truswell. 2011. *Events, Phrases and Questions*. Oxford: Oxford University Press.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of Blackbox NLP at ACL*, page pp.10.

Ethan Wilcox, Roger P. Levy, Takashi Morita, and Richard Futrell. 2018. What do rnn language models learn about filler-gap dependencies? In *Proceedings of the Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Ethan Wilcox, Roger P. Levy, Takashi Morita, and Richard Futrell. 2019b. What syntactic structures block dependencies in RNN language models? In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society (CogSci)*.

Probing RNN Encoder-Decoder Generalization of Subregular Functions using Reduplication

Max Nelson
Department of Linguistics
University of Massachusetts Amherst
manelson@umass.edu

Hossep Dolatian
Department of Linguistics
Institute for Advanced Computational Science
Stony Brook University
hossep.dolatian@stonybrook.edu

Jonathan Rawski
Department of Linguistics
Institute for Advanced Computational Science
Stony Brook University
jonathan.rawski@stonybrook.edu

Brandon Prickett
Department of Linguistics
University of Massachusetts Amherst
bprickett@umass.edu

Abstract

This paper examines the generalization abilities of encoder-decoder networks on a class of subregular functions characteristic of natural language reduplication. We find that, for the simulations we run, attention is a necessary and sufficient mechanism for learning generalizable reduplication. We examine attention alignment to connect RNN computation to a class of 2-way transducers.

1 Introduction

Reduplication is a cross-linguistically common morphological process (Moravcsik, 1978; Rubino, 2005). It is estimated that total reduplication and partial reduplication occur in 85% and 75% of the world's languages, respectively (Rubino, 2013). Total reduplication places no bound on the size of the reduplicant while partial does.

1. (a) wanita → wanita∼wanita (Indonesian)
   'woman → women'
   (b) guyon → gu∼guyon (Sundanese)
   'to jest → to jest repeatedly'

Morphological and phonological processes are sufficiently characterized by the regular class of languages and functions, and effectively computed by finite-state transducers (FSTs) (Johnson, 1972; Kaplan and Kay, 1994; Koskenniemi, 1984; Roark and Sproat, 2007). In finite-state calculus, an FST can process the input string either once in one direction (1-way FST), or multiple times by going back and forth (2-way FST). 1-way FSTs compute *rational* functions, while 2-way FSTs are more expressive, computing *regular* functions (Engelfriet and Hoogeboom, 2001; Filiot and Reynier, 2016).[1] Most morphological and phonological processes are in fact restricted to subclasses of rational functions and their corresponding 1-way FSTs (Chandlee, 2014, 2017; Chandlee and Heinz, 2018). The exception is total reduplication, which is uncomputable by 1-way FSTs due to its unboundedness (Culy, 1985; Sproat, 1992). It needs the power of 2-way FSTs, and requires subclasses of the regular functions (Dolatian and Heinz, 2018b).

This paper uses these subregular functions that characterize reduplication to probe the learning and generalization capacities of Recurrent Neural Network (RNN) architectures. While given infinite computational power, RNNs can simulate Turing machines (Siegelmann, 2012), many RNN classes and their gating mechanisms are actually expressively equivalent to weighted finite-state acceptors (Rabusseau et al., 2019; Peng et al., 2018). Furthermore, growing evidence suggests that RNNs and other sequential networks practically function as subregular automata (Merrill, 2019; Weiss et al., 2018).

We extend these subregular characterizations to

[1] In the French literature on formal language theory, 1-way FSTs compute *rational functions*. In contrast, most work in American computer science calls this class the *regular functions*. We follow French conventions because we also discuss 2-way FSTs which compute *regular functions* in their system.

test encoder-decoder (ED; Sutskever et al., 2014) networks. We use a typology of reduplication patterns computed by subregular 2-way FSTs (Dolatian and Heinz, 2019) to probe the ability of the networks to learn patterns of varying complexity. Our results suggest that when adding attention (Bahdanau et al., 2014) to these models, not only do they successfully learn and generalize all of the attested reduplication patterns that we test, but the attention acts in an alignment suggestive of the subregular 2-way FSTs. In contrast, lack of attention prohibits learning of the functions, and the generalization is suggestive of 1-way FSTs. This provides a principled glimpse into the interpretability of these networks on well-understood computational grounds, motivated by linguistic insight (Rawski and Heinz, 2019).

The paper proceeds as follows. §2 overviews the computation and learnability of reduplication. Methods, results, and discussion are in §3,§4,§5, respectively. Conclusions are in §6.

2 Background

2.1 Computing reduplication

As stated, reduplication is characterized by different subclasses of regular functions and computed by their corresponding FSTs, forming the hierarchy shown in Figure 1. 1-way FSTs compute *rational* functions. They are widely used in computational linguistics and NLP (Roche and Schabes, 1997; Beesley and Karttunen, 2003; Roark and Sproat, 2007). 2-way FSTs are more powerful. They exactly compute *regular functions*, which mathematically correspond to string-to-string transductions using Monadic Second Order logic (Engelfriet and Hoogeboom, 2001), making them the functional counterpart of the regular languages (Büchi, 1960). They have mostly been used outside of NLP (Alur and Černý, 2011).

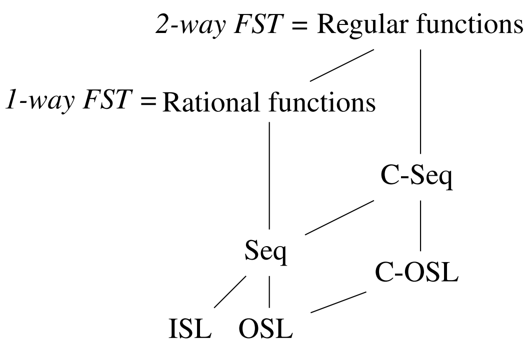


Figure 1: Hierarchy of subregular functions

When defined over a 1-way FST, all partial reduplicative functions are computable by Subsequential (Seq) functions (Chandlee and Heinz, 2012; Chandlee, 2017), which are computed by *deterministic* 1-way FSTs. Total reduplication is uncomputable by 1-way FSTs because there is no bound on the size of the reduplicant (Culy, 1985), so its output language is at least Mildly Context-Sensitive (Seki et al., 1991, 1993).

Over 2-way FSTs, both partial and total reduplication can be alternatively computed by a concatenation of subclasses of regular functions that are analogous to 1-way FST subclasses.[2] Almost all reduplicative processes, including total reduplication, are computed by Concatenated-Sequential (C-Seq) functions, which are concatenations of Seq functions (Dolatian and Heinz, 2018a,b). Most reduplication processes are sufficiently characterized by C-Seq functions because they can almost always be decomposed into two concatenated Seq functions: one to produce the reduplicant via truncation *Trunc(x)*, and one to produce an identical copy of the base *ID(x)*. Figure 2 shows such a division of a reduplicated word *gu∼guyon* (1b). [3] Figure 2 shows this division of a reduplicated word *gu∼guyon* (1b).

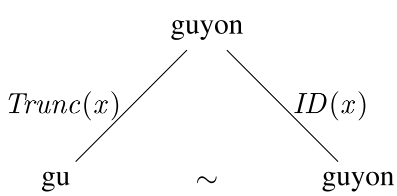


Figure 2: Initial-CV reduplication as a concatenation of subsequential functions.

Seq functions as 1-way FSTs and C-Seq functions as 2-way FSTs both compute partial reduplication, but differ in their *origin semantics* (Dolatian and Heinz, 2018b), the finite-state analog to alignment (Bojańczyk, 2014). Consider a function f, an FST T which computes f, and an input-output pair (x, y) such that $f(x) = y$. Given some substring y_j in y, the origin information of y_j with respect to T is the position x_i in x such that the

[2] See Alur et al. (2014) on the use of concatenation as a function combinator.

[3] Chandlee (2017) and Dolatian and Heinz (2018a)'s results are actually stronger. Over 1-way FSTs, most partial reduplicative processes are Input-Strictly Local (ISL) functions, a subclass of Seq functions. Over 2-way FSTs, most reduplicative processes are the concatenation of Output-Strictly Local (C-OSL) functions, a subclass of C-Seq.

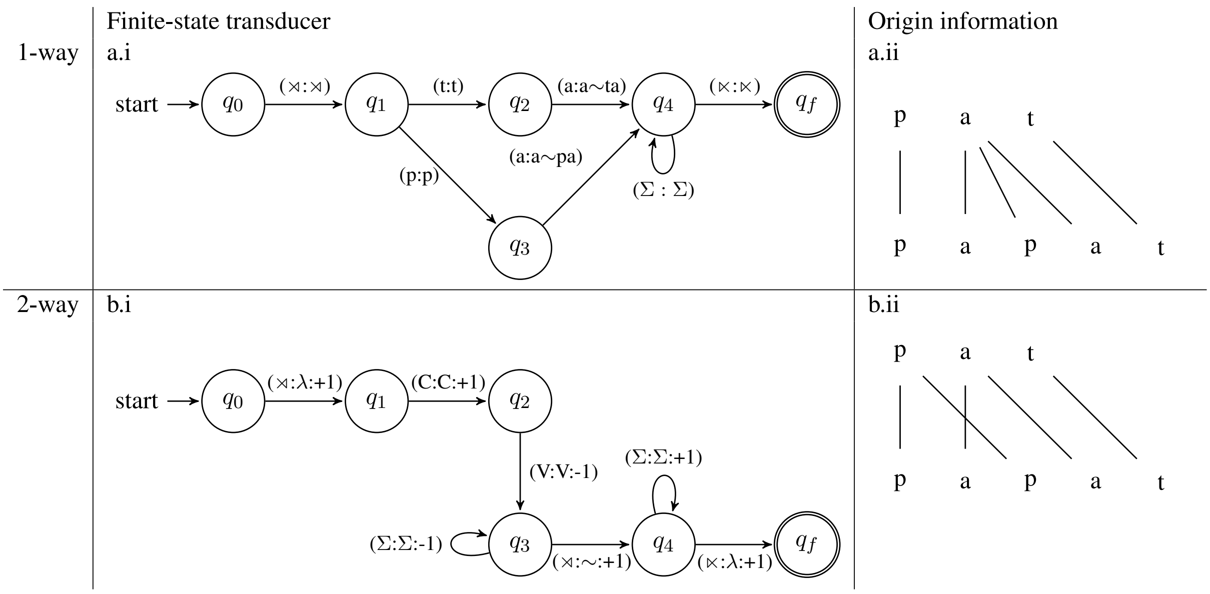


Figure 3: FSTs and origin information for initial-CV reduplication

FST's input-read head is in position x_i of the input x when the FST outputs the substring y_j.

To illustrate, consider initial-CV copying: $f(pat) = papat$. This function is computable by either the 1-way FST in Figure 3.a.i or the 2-way FST in Figure 3.b.i. The input is flanked by the end boundaries $\rtimes$,$\ltimes$. The 1-way FST implicitly advances from left-to-right on the input string. The 2-way FST advances left-to-right via the explicit +1 direction parameter until it produces the first CV string (=the reduplicant). After that, it moves right-to-left via the -1 direction parameter and reaches the start boundary $\rtimes$. It then advances left-to-right and outputs the base.[4] For the input-output pair $(pat, papat)$, the 1-way FST generates an 'alignment' or origin information such that the entire second copy '*pa*' is associated or generated from the vowel '*a*' in the input (Figure 3.a.ii). In contrast, the 2-way FST generates the alignment in Figure 3.b.ii where the second output '*p*' is associated with the input consonant '*p*'. The role of origin semantics and alignment acts as a diagnostic for understanding whether the neural networks we probe behave more like a 1-way or 2-way FST.

2.2 Learning reduplication

Chandlee et al. (2015) and Dolatian and Heinz (2018a) respectively show that ISL (Seq) and C-OSL (C-Seq) reduplicative processes are provably learnable by inducing their corresponding 1-way or 2-way FSTs in polynomial time and data. For Dolatian and Heinz (2018a), their proof relies on making the training data 'boundary enriched' with the reduplicative boundary symbol $\sim$, e.g. the training data for initial-CV reduplication is {(pat, pa∼pat), (mara, ma∼mara), etc.}. They hypothesize that learning without the boundary $\sim$ is tantamount to learning morpheme segmentation.

Gasser (1993) used simple RNNs to model reduplication and copying functions, finding that they could not properly learn reduplicative patterns. However, Prickett et al. (2018) found that ED networks, a class of RNNs that have performed well on a number of other morphological tasks (Cotterell et al., 2016; Kirov and Cotterell, 2018) could learn simple reduplicative patterns. These patterns used training data that did not represent a realistic language learning scenario, since all words had the same length and syllables were limited to a CV structure. We test the extent to which ED networks are capable of learning more realistic reduplicative functions. We find that vanilla EDs, like Prickett et al.'s, struggle to scale to realistic data, while EDs augmented with an attention mechanism easily acquire complex, natural-language-based reduplication patterns.

3 Methods

3.1 Data

We use a library of C-Seq transducers derived from the typology of natural language reduplication patterns (Dolatian and Heinz, 2019) to generate sets of input-output mappings which we use to

[4] See the appendix for more details on 2-way FSTs.

query several ED architectures.

The typology exhibits multiple parameters and distinctions. Already mentioned was the distinction between partial and total reduplication: copying a bounded substring of the input *gu∼guyon* (1b) vs. copying the entire potentially unbounded input *wanita* → *wanita∼wanita* (1a).

For partial reduplication, one subparameter is whether the reduplicant has a fixed size or a variable size that is still smaller than some fixed natural number. Fixed-sized partial reduplication is the most common pattern, e.g. initial CV-copying: *gu∼guyon* (1b) (Moravcsik, 1978; Rubino, 2005). One instantiation of variable-length partial reduplication is copying the initial foot (2(a)i) (Marantz, 1982), or syllable (2(b)i) (Haugen, 2005), which used to be unattested (Moravcsik, 1978). Another subparameter is whether the reduplicant is adjacent to the segments it copied (1b) or non-adjacent, i.e. wrong-sided (2c). Wrong-sided reduplication is controversial (Nelson, 2003) but attested (Riggle, 2004).

2. (a) i. (dimu)rU → dimu∼dimurU (Yidin)
   ‘house’ → ‘houses’
   ii. (gindal)ba → gindal∼gindalba
   ‘lizard sp.’ → ‘lizards’
   (b) i. vu.sa → vu∼vusa (Yaqui)
   ‘awaken’ → ‘awaken (habitual)’
   ii. vam.se → vam∼vamse
   ‘hurry’ → ‘hurry (habitual)’
   (c) qanga → qanga∼qan (Koryak)
   ‘fire’ → ‘fire (absolute)’

Over 1-way FSTs, adjacent partial reduplication and foot/syllable copying are ISL while wrong-sided reduplication is Seq. Over 2-way FSTs, total reduplication and all the above partial reduplication functions are C-OSL, a subclass of C-Seq.[5]

We tested multiple patterns, including partial initial and wrong-sided reduplication of the first two syllables, total reduplication, and partial initial reduplication of the first two segments. For each pattern, the models are given base strings as input and trained to reproduce the base string along with its reduplicant (i.e. a right or left concatenated fully or partially copied form). For all patterns, 10,000 input-output pairs are generated, 7,000 of which are used to train the models while the remaining 3,000 are held out to test model generalization. For clarity the ∼ symbol is used throughout this paper to denote the boundary between a base and its reduplicant, however no such boundary is present in the model’s training data.

3.2 Models

Many ED networks were built and trained on the datasets described above. EDs are composed of a recurrent encoder, which sequentially processes an input string to yield a vector representation of the sequence in $\mathbb{R}^n$, and a recurrent decoder which takes the encoded representation of the input as a starting state and continues producing outputs until it produces a target stop symbol or reaches an experimenter-defined maximum length. The use of recurrent layers in both in the encoder and decoder allows EDs to map variable-length input sequences to variable-length output sequences, with no necessary relationship between the length of the input and target output (Sutskever et al., 2014).

Simple (SRNN) and gated (GRU) recurrence relations were tested as the encoder and decoder recurrent layers.[6] In SRNN layers the network’s state at any timepoint, h_t, is dependent only on the input at that timepoint and the network’s state at the previous timepoint (Elman, 1990).

$$h_t = tanh(W_x x_t + b_{ih} + W_h h_{t-1} + b_{hh}) \quad (1)$$

Consequently, in an SRNN there is only one path for the forward and backward propogation of information. This leads to potential problems for SRNNs in representing long-distance dependencies (Bengio et al., 1994) and problems with the backward flow of information during training (Hochreiter et al., 2001). GRU layers have a series of gates, called the reset r_t, update z_t, and new n_t gates, which create an alternative path of information flow (Cho et al., 2014), as shown in (2).

$$\begin{aligned} r_t &= \sigma(W_{ir} x_t + b_{ir} + W_{hr} h_{t-1} + b_{hr}) \\ z_t &= \sigma(W_{iz} x_t + b_{iz} + W_{hz} h_{t-1} + b_{hz}) \\ n_t &= tanh(W_{in} x_t + b_{in} + r_t \odot (W_{hn} h_{t-1} + b_{hn})) \\ h_t &= (1 - z_t) \odot n_t + z_t \odot h_{t-1} \end{aligned} \quad (2)$$

In a classic ED architecture, the encoded representation of the input is the only piece of infor-

[5] Foot and syllable copying are C-OSL if the input is marked by syllable/foot boundaries; otherwise they’re C-Seq.

[6] GRU layers have been shown to behave comparably to LSTMs, despite having fewer parameters (Chung et al., 2014). One difference between GRU and LSTM comes from (Weiss et al., 2018), who suggests that LSTMs are able to learn arbitrary $a^n b^n$ patterns while GRUs are not.

mation that is passed from the encoder to the decoder. This forces all necessary information in the input to be stored in this vector and preserved throughout the decoding process. In all experiments presented below, the target outputs consist of a concatenated reduplicant and base. Because the model must reproduce the base. it must preserve the identity of all phonemes in the input sequence. In order to test the ability of the model to learn the reduplicative function independent of its ability to store segment identities over arbitrarily long spans, a global weighted attention mechanism was incorporated into some of the models. This is a key point of departure from previous attempts to model reduplication with ED networks.

Attention allows the decoder to selectively attend to the hidden states of the encoder by learning a set of weights, W_{att}, which map the decoder's current state to a set of weights over timesteps in the input, and then concatenating the current decoder hidden state, h_t, the weighted combination of all encoder hidden states to yield a new current decoder state, h_{tt} (Bahdanau et al., 2014; Luong et al., 2015). This is illustrated in Equation 3, where E is a matrix of size *input length* $\times$ *hidden dimensionality* such that the ith row contains the encoder hidden state at timepoint i.

$$h_{tt} = \text{CAT}(h_t, \sigma(W_{att}h_t)^T E) \tag{3}$$

In this way, the decoder can pull information directly from the encoder by learning an alignment between the output and input representations.

The next section presents the results of training networks with either SRNN or GRU recurrent layers with and without an attention mechanism and then testing their ability to generalize the target pattern. All networks are trained to minimize phoneme level cross-entropy.

4 Results

In this section, we test ED networks on their ability to learn partial (§4.1,4.3) and total reduplication (4.2). Within partial reduplication, we test if they can learn adjacent reduplication vs. wrong-sided reduplication, and fixed-size vs. variable-length reduplication.

4.1 Partial reduplication

One simplifying assumption of previous work is that the reduplicant is a fixed-length substring of the base. This section tests the extent to which ED networks are able to learn reduplicative functions that copy a *variably* sized substring of the base in a way that is sensitive to linguistic structure which is not explicitly encoded in the training data.

Models were trained on initial and wrong-sided reduplication in which the reduplicant consisted of the first two-syllables in the word. Syllables were defined to be as onset-maximizing as possible and complex onsets and codas were included in the training data. This means that, for words with more than two syllables, the target reduplicant included everything between the left edge of the word and the right edge of the second vowel (initial: tasgatri→tasga∼tasgatri, wrong-sided: tasgatri→tasgatri∼tasgat). For words with only one or two vowels the reduplicant was the entire word (tasgat→tasgat∼tasgat). Due to the variable presence of onsets and codas, both simple and complex, reduplicants in these test cases vary in length between 2 and 10 phonemes, and may contain either 1 or 2 vowels.

In order for the model to learn this pattern, it must learn to identify which phonemes are consonants and which are vowels, must learn the syllabification rules, and must learn to handle the one-syllable exceptional case. Table (1) shows the generalization accuracy for the tested network architectures on datasets instantiating this pattern. As will be discussed in §4.3, the success of networks without attention is partially dependent on characteristics of the target language, namely the size of the language's segment inventory and permitted string lengths. To highlight these effects, results are reported from a representative *small* language, which has 10 unique phonemes and permits bases of between 3 and 9 segments, and a *large* language, which has 26 unique phonemes and permits bases of between 3 and 15 segments.

		Non-attention		Attention	
		Small	Large	Small	Large
Initial	SRNN	0.107	0.000	0.997	0.990
	GRU	0.787	0.234	1.000	1.000
wrong-sided	SRNN	0.001	0.000	0.995	0.994
	GRU	0.682	0.236	1.000	1.000

Table 1: Generalization accuracy by network type for all four languages that were tested.

The results suggest that the attention-based models are able to learn and generalize both initial and wrong-sided two-syllable reduplication patterns in a way that is robust to recurrence relation and language size. Non-attention GRU models show mild success in the small language, but seem heavily affected by language size, a result that will be explored thoroughly in §4.3. Non-attention RNN models are unable to learn the patterns in any of the simulations we ran.

The attention-based models are able to learn an alignment between the input and output that allows them to pull information directly from the input during decoding, sidestepping a potential information bottleneck at the encoded representation. To illustrate the alignment functions, an SRNN trained on two-syllable initial reduplication was used to make predictions about novel forms and the attention weights were stored. Figure (4) plots the attention weights for this model at every step in decoding for the three-syllable word *pastapo* and the two-syllable word *spaftof* (‘<’ and ‘>’ represent start-of-sequence and end-of-sequence tokens, respectively).

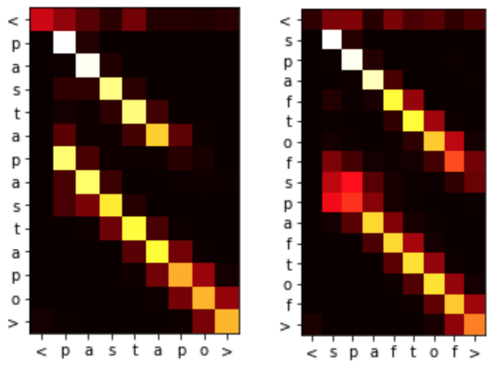


Figure 4: Attention weights over input (horizontal) at each time step of correct decoding of reduplicated form (vertical) for two-syllable initial reduplication of the words *pastapo* and *spaftof*. Darker squares indicate a lower weight on the alignment between two timesteps.

The attention weights confirm that the model learned an alignment between corresponding phonemes in the input and output. A single phoneme in the input has an output correspondent in both the base and reduplicant. These examples also illustrate the model's ability to i) identify the cut-off point for the reduplicant even when it is not explicitly marked and to ii) identify exceptional cases where the word is only two syllables and thus the reduplicant consists of material past the second vowel. In *pastapo* the model cuts off the reduplicant after the second vowel and in *spaftof* the model correctly includes the coda consonant because the word consists of only two syllables.

This section showed that attention-based models can learn initial and wrong-sided reduplication even when the pattern is complicated by sensitivity to linguistic structure that results in variable-length reduplicants. Once the network has learned enough structure to perform syllabification, the two-syllable *partial* reduplicative function is C-Seq. The next section examines the extent to which these networks learn *unbounded* copying, i.e. total reduplication.

4.2 Total reduplication

We test the ability of ED networks to learn and generalize total reduplication: *wanita* → wanita∼wanita (1a). As mentioned, total reduplication is not a rational function and is uncomputable with a 1-way FST, since there is no upper bound on the size of the copied string. However, it *is* a C-Seq function and computable by the corresponding 2-way FST. Total reduplication is thus a crucial test case for the RNN behavior.

As in §4.1, SRNN and GRU models with and without attention are trained on large and small languages where small languages have 10 phonemes and base lengths between 3 and 9 segments, and large languages have 26 phonemes and base lengths between 3 and 15 segments.

	Non-attention		Attention	
	Small	Large	Small	Large
SRNN	0.046	0.0	0.999	0.985
GRU	0.705	0.211	0.999	0.995

Table 2: Generalization accuracy by network type on both the large and small total reduplication patterns.

Table 2 shows the generalization accuracy for all network configurations. The results are nearly identical to those for the partial reduplication patterns in §4.1. Attention models can robustly learn the pattern, with negligible effects of recurrence relation or language size. Without attention, no model fully succeeds in generalizing the total reduplication pattern, with the best performance coming from the GRU on the small language.

These results show that attention-based models can learn a generalizable total reduplication function as well as they can learn partial reduplication functions. This means that attention-based ED network generalization does not distinguish between total and partial reduplication, despite glaring functional and automata-theoretic differences in the functions themselves. This clearly suggests that an RNN architecture that can learn both functions necessarily computes a C-Seq function, which properly includes both processes. Furthermore, as discussed in §5, the interpretability of the corresponding FST characterization (2-way vs 1-way) and its origin semantics provides a direct computational link to the attention mechanism of these RNN architectures.

4.3 Alphabet size and string length effects

As shown so far, network architecture is not the only factor that influences a network's ability to learn a target reduplicative function. The composition of the target language, in terms of the number of segments in the language and the number of permitted string lengths, can have a dramatic effect on model behavior.

The effect of model architecture and language composition was investigated by testing the extent to which all network configurations could learn simple reduplication pattern while systematically varying the size of the segment inventory and permitted base lengths in the data. The reduplicative function chosen for these tests copied a fixed-window of two segments for initial reduplication: *guyon→gu∼guyon*. This was chosen because it is typologically well-attested (Moravcsik, 1978; Rubino, 2005, 2013) and also predicted to be the simplest reduplication pattern for the network to learn (since it is insensitive to linguistic structure and has a fixed-length reduplicant).

Data that followed this pattern was generated for languages with 10, 18, and 26 unique phonemes in their inventory and which permit bases to vary from 3 to between 5 and 10 segments. These results are shown in Figure (5).[7] The top panel shows the effect of alphabet size; string lengths are fixed between 3 and 8. The bottom panel, which shows the effect of string lengths; alphabet size is fixed at 26. The lines paralleling 1.0 in the top panel show that the ability of attention-based models to learn the target function is robust to alphabet size. The lines paralleling 1.0 in the bottom panel illustrate that attention-based models are similarly robust to string length variation.

In contrast, the non-attention models show large effects of alphabet size and string length. The non-

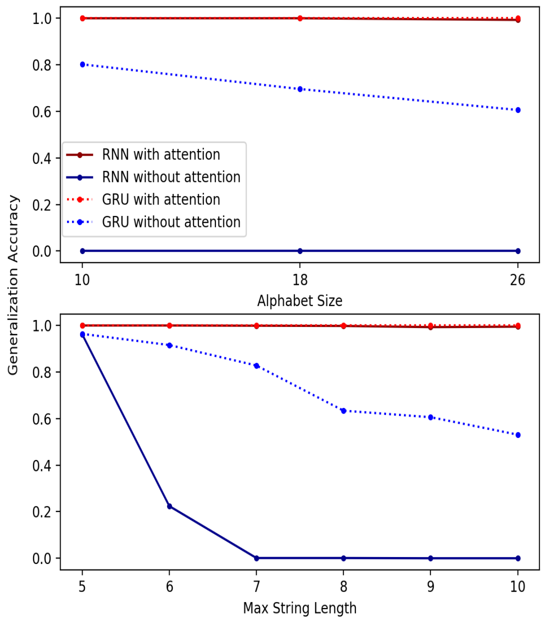


Figure 5: Effect on varying alphabet size and maximum string length, with minumum string length fixed at 3, on generalization accuracy.

attention SRNN shows very limited success. It is able to generalize with a very limited number of string lengths; but when maximum string length exceeds 7, it is no longer able to learn the target function at all. Consequently, the accuracy of the SRNN in the top panel, where maximum string length is fixed at 9, is stuck at 0.0 across all alphabet sizes.

The effects of both string length and alphabet size are also visible for the non-attention GRU. In the top panel, where maximum string length is fixed at 9, a decrease in generalization accuracy as a function of alphabet size is observed. The effect of maximum string length on the non-attention GRU is less dramatic than on the SRNN, but the GRU still displays a decrease from near ceiling accuracy with lengths between 3 and 5, to ~ 0.60 when lengths range between 3 and 10.

The sensitivity of non-attention SRNN and GRU models to alphabet size and string length are likely a result of the fact that these models are unable to directly reference the input during decoding and must pass all information through the encoder bottleneck. This hypothesis is strengthened by the fact that, without attention, the GRU performs much better than the SRNN. The GRU has extra gates between timepoints which aid in the long-distance preservation of information, mitigating the bottleneck problem to an extent. How-

[7] The reported results are from initial reduplication with a window size of two segments, however, wrong-sided reduplication and a larger window size of three were also tested with nearly identical results.

ever, while this assists the GRU network, it is not enough to make alphabet size and word length non-issues. The non-attention GRU is similar in architecture to the LSTM model of Prickett et al. (2018), with a slightly different training objective, suggesting that their model would similarly have difficulty scaling up.

The lack of a difference between the attention-based GRU and SRNN corroborates the idea that when this information bottleneck is not an issue both architectures are capable of learning generalizable reduplication.

5 Discussion

5.1 Origin semantics and alignment

As explained in §2.1, partial reduplication can be computed as a function with either 1-way or 2-way FSTs. However, the two finite-state algorithms differ in their origin semantics or alignment. The alignment difference is simulated by the attention-based RNNs. The alignments learned by attention-based models for partial reduplication in §4.1 and §4.3 are analogous to the origin semantics computed by the 2-way FST. We illustrate in Figure 6.

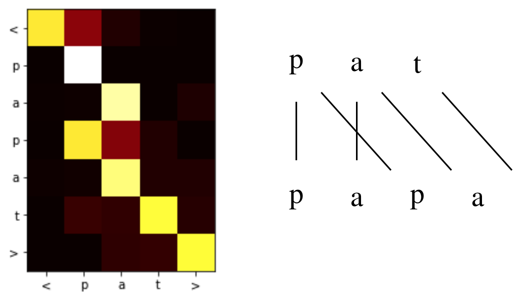


Figure 6: (left): Attention weights over input (horizontal) at each time step of correct decoding of reduplicated form (vertical) for the mapping *pat→pa∼pat*. Darker squares indicate a lower weight on the alignment between two timesteps. (right): Origin semantics of 2-way FST from Figure 3b.ii.

While both Seq and C-Seq functions sufficiently characterize partial reduplication, this 2-way-like alignment suggests that the RNNs are generalizing C-Seq functions (see Fig. 4 for other examples). This extends to total reduplication (§4.2) whose alignment when learned by the attention-based RNNs suggests the same origin information as 2-way FSTs. These results hint at the expressivity of the ED models, explicitly connecting their computations to the 2-way automata characterizing this subregular class.

5.2 Generality of copying mechanisms

The results suggest that the same general-purpose mechanism can be used to model both partial and total reduplication. The attention-based RNNs learned both processes with near-equal ease and generalizability and the same tools. This learning result fits well with reduplicative typology and theory. Partial and total reduplication are typologically and diachronically linked. If a language has partial reduplication, then it almost always has total reduplication, often because the former developed from the latter (Rubino, 2013). Because of this dependence, certain linguistic theories use the same mechanisms to generate both processes (Inkelas and Zoll, 2005).

Computationally, our result fits with the characterization of reduplication over 2-way FSTs (Dolatian and Heinz, 2018b) but not over 1-way FSTs (Chandlee et al., 2012). Because total reduplication cannot be modeled by a 1-way FSTs, some suggest that total and partial reduplication are ontologically different and should be computed with separate mechanisms (Roark and Sproat, 2007; Chandlee, 2017). In contrast, when computed over 2-way FSTs, both reduplicative processes fall under the *same* subclass of C-Seq functions.

5.3 Scaling problems

The results from §4.3 shows that attention-based RNNs could equally well learn a partial reduplication function regardless of alphabet size input size. In contrast, attention-less RNNs suffer. For an attention-less RNN, learning initial-CV copying with a small alphabet over smaller words is significantly easier then learning it with a larger alphabet over larger words. Their scaling difficulty is reminiscent of 1-way FST treatments of partial reduplication. To compute partial reduplication, 1-way FSTs can suffer a significant state explosion as alphabet size or reduplicant size increases. This is why some call 1-way FSTs 'burdensome models' for partial reduplication (Roark and Sproat, 2007, 54). 2-way FSTs do not suffer from state explosion (Dolatian and Heinz, 2018b).

6 Conclusions

We showed that RNN encoder-decoder networks with attention can learn partial and total reduplication patterns. Non-attention models exhibited mixed success in learning generalizable reduplication functions in a way that was dependent on alphabet size and string length, suggesting that their failure is attributable to the information bottleneck between encoder and decoder rather than an inability to learn the target function. This corroborates the finding by Weiss et al. (2018) that recurrent networks' expressive power is restricted in practice, and shows the fruitfulness of using well-understood subregular classes to probe this expressivity.

References

Rajeev Alur, Adam Freilich, and Mukund Raghothaman. 2014. Regular combinators for string transformations. In Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, pages 9:1–9:10, New York, NY, USA. ACM.

Rajeev Alur and Pavol Černý. 2011. Streaming transducers for algorithmic verification of single-pass list-processing programs. In Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '11, pages 599–610, New York, NY, USA. ACM.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Kenneth Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. CSLI Publications.

Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157–166.

Mikołaj Bojańczyk. 2014. Transducers with origin information. In Automata, Languages, and Programming, pages 26–37, Berlin, Heidelberg. Springer.

J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. Mathematical Logic Quarterly, 6(1-6):66–92.

Jane Chandlee. 2014. Strictly Local Phonological Processes. Ph.D. thesis, University of Delaware, Newark, DE.

Jane Chandlee. 2017. Computational locality in morphological maps. Morphology, pages 1–43.

Jane Chandlee, Angeliki Athanasopoulou, and Jeffrey Heinz. 2012. Evidence for classifying metathesis patterns as subsequential. In The Proceedings of the 29th West Coast Conference on Formal Linguistics, pages 303–309, Somerville, MA. Cascillida Press.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In 14th Meeting on the Mathematics of Language, pages 112–125.

Jane Chandlee and Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, SIGMORPHON '12, pages 42–51, Montreal, Canada. Association for Computational Linguistics.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. Linguistic Inquiry, 49(1):23–60.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared taskmorphological reinflection. In Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 10–22.

Christopher Culy. 1985. The complexity of the vocabulary of Bambara. Linguistics and philosophy, 8:345–351.

Hossep Dolatian and Jeffrey Heinz. 2018a. Learning reduplication with 2-way finite-state transducers. In Proceedings of Machine Learning Research: International Conference on Grammatical Inference, volume 93 of Proceedings of Machine Learning Research, pages 67–80, Wroclaw, Poland.

Hossep Dolatian and Jeffrey Heinz. 2018b. Modeling reduplication with 2-way finite-state transducers. In Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Brussells, Belgium. Association for Computational Linguistics.

Hossep Dolatian and Jeffrey Heinz. 2019. Redtyp: A database of reduplication with computational models. In Proceedings of the Society for Computation in Linguistics, volume 2. Article 3.

Jeffrey L Elman. 1990. Finding structure in time. Cognitive science, 14(2):179–211.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. ACM Trans. Comput. Logic, 2(2):216–254.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. ACM SIGLOG News, 3(3):4–19.

Michael Gasser. 1993. Learning words in time: Towards a modular connectionist account of the acquisition of receptive morphology. Indiana University, Department of Computer Science.

Jason Haugen. 2005. Reduplicative allomorphy and language prehistory in Uto-Aztecan. In Bernhard Hurch, editor, Studies on reduplication, 28, pages 315–350. Walter de Gruyter, Berlin.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In John F Kolen and Stefan C Kremer, editors, A field guide to dynamical recurrent networks. John Wiley & Sons.

Sharon Inkelas and Cheryl Zoll. 2005. Reduplication: Doubling in Morphology. Cambridge University Press, Cambridge.

C Douglas Johnson. 1972. Formal aspects of phonological description. Mouton The Hague.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. Computational linguistics, 20(3):331–378.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. Transactions of the Association for Computational Linguistics, 6:651–665.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In Proceedings of the 10th international conference on Computational Linguistics, pages 178–181. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025.

Alec Marantz. 1982. Re reduplication. Linguistic inquiry, 13(3):435–482.

William Merrill. 2019. Sequential neural networks as automata. In Proceedings of the Deep Learning and Formal Languages workshop at ACL 2019.

Edith Moravcsik. 1978. Reduplicative constructions. In Joseph Greenberg, editor, Universals of Human Language, volume 1, pages 297–334. Stanford University Press, Stanford, California.

Nicole Alice Nelson. 2003. Asymmetric anchoring. Ph.D. thesis, Rutgers University, New Brunswick, NJ.

Hao Peng, Roy Schwartz, Sam Thomson, and Noah A Smith. 2018. Rational recurrences. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1203–1214.

Brandon Prickett, Aaron Traylor, and Joe Pater. 2018. Seq2seq models with dropout can learn generalizable reduplication. In Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 93–100.

Guillaume Rabusseau, Tianyu Li, and Doina Precup. 2019. Connecting weighted automata and recurrent neural networks through spectral learning. In AISTATS.

Jonathan Rawski and Jeffrey Heinz. 2019. No free lunch in linguistics or machine learning: Response to pater. Language, 94:1.

Jason Riggle. 2004. Nonlocal reduplication. In Proceedings of the 34th meeting of the North Eastern Einguistics Society. Graduate Linguistic Student Association, University of Massachusetts.

Brian Roark and Richard Sproat. 2007. Computational Approaches to Morphology and Syntax. Oxford University Press, Oxford.

Emmanuel Roche and Yves Schabes. 1997. Finite-state language processing. MIT press.

Carl Rubino. 2005. Reduplication: Form, function and distribution. In Studies on reduplication, pages 11–29. Mouton de Gruyter, Berlin.

Carl Rubino. 2013. Reduplication. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. Theoretical Computer Science, 88(2):191–229.

Hiroyuki Seki, Ryuichi Nakanishi, Yuichi Kaji, Sachiko Ando, and Tadao Kasami. 1993. Parallel multiple context-free grammars, finite-state translation systems, and polynomial-time recognizable subclasses of lexical-functional grammars. In Proceedings of the 31st annual meeting on Association for Computational Linguistics, pages 130–139. Association for Computational Linguistics.

Hava T Siegelmann. 2012. Neural networks and analog computation: beyond the Turing limit. Springer Science & Business Media.

Richard William Sproat. 1992. Morphology and computation. MIT press, Cambridge:MA.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 740–745.

A Appendix

The definition and illustration for 2-way FSTs are taken from Dolatian and Heinz (2018b). We use $\rtimes, \ltimes$ as the start and end boundaries.

3) **Definition:** A 2-way, deterministic FST is a six-tuple $(Q, \Sigma_{\ltimes}, \Gamma, q_0, F, \delta)$ such that:

   Q is a finite set of states,
   $\Sigma_{\ltimes} = \Sigma \cup \{\rtimes, \ltimes\}$ is the input alphabet,
   Γ is the output alphabet,
   $q_0 \in Q$ is the initial state,
   $F \subseteq Q$ is the set of final states,
   $\delta : Q \times \Sigma \rightarrow Q \times \Gamma^* \times D$ is the transition function where the direction $D = \{-1, 0, +1\}$.

For a survey on legitimate configurations in a 2-way FSTs, its computational properties, and complexity diagnostics, please see Dolatian and Heinz (2018b).

To illustrate 2-way FSTs, Figure 7 shows a 2-way FST for total reduplication. The 2-way operates by:

1. reading the input tape once from left to right in order to output the first copy,
2. going back to the start of the input tape by moving left until the start boundary $\rtimes$ is reached,
3. reading the input tape once more from left to right in order to output the second copy.

Specifically, this figure is interpreted as follows. The symbol Σ stands for any segment in the alphabet except for $\{\rtimes, \ltimes\}$. The arrow from q_1 to itself means this 2-way FST reads Σ, writes Σ, and advances the read head one step to the right on the input tape. The boundary symbol $\sim$ is a symbol in the output alphabet Γ, and is not necessary. We include it only for illustration.

We show an example derivation in Figure 8 for the input-output pair (*wanita*, *wanita*$\sim$*wanita*) (1a using the 2-way FST in Figure 7. The derivation shows the configurations of the computation for the input *wanita* and is step by step. Each tuple consists of four parts: *input string, output string, current state, transition*. In the *input string*, we underline the input symbol which FST will read next. The *output string* is what the 2-way FST has outputted up to that point. The symbol λ marks the empty string. The *current state* is what state the FST is currently in. The *transition* represents the used transition arc from input to output. In the first tuple, there is no transition arc used (N/A). But for other tuples, the form of the arc is:

$$\textit{input state} \xrightarrow[\text{direction}]{\text{input symbol:output string}} \textit{output state}$$

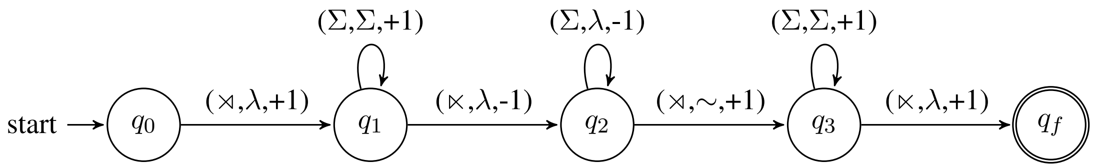


Figure 7: 2-way FST for total reduplication.

Outputting the first copy									
1.	($\underline{\rtimes}$wanita$\ltimes$,	λ,	q_0 ,	N/A)	9.	($\rtimes$wanit$\underline{\text{a}}$$\ltimes$,	wanita$\sim$,	q_2,	$q_1 \xrightarrow[-1]{\ltimes:\sim} q_2$)
2.	($\rtimes$$\underline{\text{w}}$anita$\ltimes$,	λ,	q_1,	$q_0 \xrightarrow[+1]{\rtimes:\lambda} q_1$)	10.	($\rtimes$wani$\underline{\text{t}}$a$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
3.	($\rtimes$w$\underline{\text{a}}$nita$\ltimes$,	w,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)	11.	($\rtimes$wan$\underline{\text{i}}$ta$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
4.	($\rtimes$wa$\underline{\text{n}}$ita$\ltimes$,	wa,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)	12.	($\rtimes$wa$\underline{\text{n}}$ita$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
5.	($\rtimes$wan$\underline{\text{i}}$ta$\ltimes$,	wan,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)	13.	($\rtimes$w$\underline{\text{a}}$nita$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
6.	($\rtimes$wani$\underline{\text{t}}$a$\ltimes$,	wani,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)	14.	($\rtimes$$\underline{\text{w}}$anita$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
7.	($\rtimes$wanit$\underline{\text{a}}$$\ltimes$,	wanit,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)	11.	($\underline{\rtimes}$wanita$\ltimes$,	wanita$\sim$,	q_2,	$q_2 \xrightarrow[-1]{\Sigma:\lambda} q_2$)
8.	($\rtimes$wanita$\underline{\ltimes}$,	wanita,	q_1,	$q_1 \xrightarrow[+1]{\Sigma:\Sigma} q_1$)					
Outputting the second copy									
12.	($\rtimes$$\underline{\text{w}}$anita$\ltimes$,	wanita$\sim$,	q_3,	$q_2 \xrightarrow[+1]{\rtimes:\lambda} q_3$)	15.	($\rtimes$wani$\underline{\text{t}}$a$\ltimes$,	wanita$\sim$wani,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)
13.	($\rtimes$w$\underline{\text{a}}$nita$\ltimes$,	wanita$\sim$w,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)	15.	($\rtimes$wanit$\underline{\text{a}}$$\ltimes$,	wanita$\sim$wanit,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)
14.	($\rtimes$wa$\underline{\text{n}}$ita$\ltimes$,	wanita$\sim$wa,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)	16.	($\rtimes$wanita$\underline{\ltimes}$,	wanita$\sim$wanita,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)
14.	($\rtimes$wan$\underline{\text{i}}$ta$\ltimes$,	wanita$\sim$wan,	q_3,	$q_3 \xrightarrow[+1]{\Sigma:\Sigma} q_3$)	17.	($\rtimes$wanita$\ltimes$,	wanita$\sim$wanita,	q_f,	$q_3 \xrightarrow[+1]{\ltimes:\ltimes} q_f$)

Figure 8: Derivation of *wanita*$\rightarrow$*wanita*$\sim$*wanita*.

Where New Words Are Born: Distributional Semantic Analysis of Neologisms and Their Semantic Neighborhoods

Maria Ryskina[1] **Ella Rabinovich**[2] **Taylor Berg-Kirkpatrick**[3]
David R. Mortensen[1] **Yulia Tsvetkov**[1]

[1]Language Technologies Institute, Carnegie Mellon University,
{mryskina,dmortens,ytsvetko}@cs.cmu.edu
[2]Department of Computer Science, University of Toronto, ella@cs.toronto.edu
[3]Computer Science and Engineering, University of California, San Diego, tberg@eng.ucsd.edu

Abstract

We perform statistical analysis of the phenomenon of *neology*, the process by which new words emerge in a language, using large diachronic corpora of English. We investigate the importance of two factors, semantic sparsity and frequency growth rates of semantic neighbors, formalized in the distributional semantics paradigm. We show that both factors are predictive of word emergence although we find more support for the latter hypothesis. Besides presenting a new linguistic application of distributional semantics, this study tackles the linguistic question of the role of language-internal factors (in our case, sparsity) in language change motivated by language-external factors (reflected in frequency growth).[1]

1 Introduction

Natural languages are constantly changing as the context of their users changes (Aitchison, 2001). Perhaps the most obvious type of change is the introduction of new lexical items, or *neologisms* (a process called "neology"). Neologisms have various sources. They are occassionally coined out of whole cloth (*grok*). More frequently, they are loanwords from another language (*tahini*), derived words (*unfriend*), or existing words that have acquired new senses (as when *web* came to mean 'World Wide Web' and then 'the Internet'). While neology has long been of interest to linguists (§2), there have been relatively few attempts to study it as a global, systemic phenomenon. Computational modeling and analysis of neology is the focus of our work.

What are the factors that predict neology? Certainly, social context plays a role. Close interaction between two cultures, for example, may result in increased borrowing (Appel and Muysken, 2006). We hypothesize, though, that there are other factors involved—factors that can be modeled more directly. These factors can be understood in terms of **supply** and **demand**.

Bréal (1904) introduced the idea that the distribution of words in semantic space tends towards uniformity. This framework predicts that new words would emerge where they would repair uniformity—where there was a space not occupied by a word. This could be viewed as supply-driven neology. Next, demand plays a role as well as supply (Campbell, 2013): new words emerge in "stylish" neighborhoods, corresponding to domains of discourse that are increasing in importance (reflected by the increasing frequency of the words in those neighborhoods).

We operationalize these ideas using distributional semantics (Lenci, 2018). To formalize the hypothesis of supply-driven neology for computational analysis, we measure **sparsity of areas in the word embedding space** where neologisms would later emerge. The demand-driven view of neology motivates our second hypothesis: **neighborhoods in the embedding space containing words rapidly growing in frequency** are more likely to produce neologisms. Both hypotheses are defined more formally in §3.

Having formalized our hypotheses in terms of word embeddings, we test them by comparing the distributions of the corresponding metrics for a set of automatically identified neologisms and a control set. Methodology of the word selection and hypothesis testing is detailed in §4. We discuss the results in §5, demonstrating evidence for both hypotheses, although the demand-driven hypothesis has more significant support.

2 Background

Neology Specific sources of neologisms have been studied: lexical borrowing (Taylor and Grant,

[1]The code and word lists are available at https://github.com/ryskina/neology

2014; Daulton, 2012), morphological derivation (Lieber, 2017), blends or portmanteaus (Cook, 2012; Renner et al., 2012), clippings, acronyms, analogical coinages, and arbitrary coinages, but these studies have tended to look at neologisms atomistically, or to explicate the social conditions under which a new word entered a language rather than looking at neologisms in systemic context.

To address this deficit, we look back to the seminal work of Michel Bréal, who introduced the idea that words exist in a semantic space. His work implies that, other things being equal, the semantic distribution of words tends towards uniformity (Bréal, 1904). This is most explicit in his law of differentiation, which states that near synonyms move apart in semantic space, but has other implications as well. For example, this principle predicts that new words are more likely to emerge where they would increase uniformity. This could be viewed as supply-driven neology—new words appear to fill gaps in semantic space (to express concepts that are not currently lexicalized).

In linguistic literature neology is often associated with new concepts or domains of increasing importance (Campbell, 2013). Just as there are factors that predict where houses are built other than the availability of land, there are factors that predict where new words emerge other than the availability of semantic space. Demand, we hypothesize, plays a role as well as supply.

Most existing computational research on the mechanisms of neology focuses on discovering sociolinguistic factors that predict acceptance of emerging words into the mainstream language and growth of their usage, typically in online social communities (Del Tredici and Fernández, 2018). The sociolinguistic factors can include geography (Eisenstein, 2017), user demographics (Eisenstein et al., 2012, 2014), diversity of linguistic contexts (Stewart and Eisenstein, 2018) or word form (Kershaw et al., 2016). To the best of our knowledge, there is no prior work focused on discovering factors predictive of the emergence of new words rather than modeling their lifecycle. We model language-external processes indirectly through their reflection in language, thereby capturing phenomena evident of our hypotheses through linguistic analysis.

Distributional semantics and language change Word embeddings have been successfully used for different applications of the diachronic analysis of language (Tahmasebi et al., 2018). The closest task to ours is analyzing meaning shift (tracking changes in word sense or emergence of new senses) by comparing word embedding spaces across time periods (Kulkarni et al., 2015; Xu and Kemp, 2015; Hamilton et al., 2016; Kutuzov et al., 2018). Typically, embeddings are learned for discrete time periods and then aligned (but see Bamler and Mandt, 2017). There has also been work on revising the existing methodology, specifically accounting for frequency effects in embeddings when modeling semantic shift (Dubossarsky et al., 2017).

Other related questions where distributional semantics proved useful were exploring the evolution of bias (Garg et al., 2018) and the degradation of age- and gender-predictive language models (Jaidka et al., 2018).

3 Hypotheses

This section outlines the two hypotheses we introduced earlier from the linguistic perspective, formalized in terms of distributional semantics.

Hypothesis 1 *Neologisms are more likely to emerge in sparser areas of the semantic space.* This corresponds to the supply-driven neology hypothesis: we assume that areas of the space that contain fewer semantically related words are likely to give birth to new ones so as to fill in the 'semantic gaps'. Word embeddings give us a natural way of formalizing this: since semantically related words have been shown to populate the same regions in embeddings spaces, we can approximate semantic sparsity (or density) of a word's neighborhood as the number of word vectors within a certain distance of its embedding.

Hypothesis 2 *Neologisms are more likely to emerge in semantic neighborhoods of growing popularity.* Here we formalize our demand-driven view of neology, which assumes that growing frequency of words in a semantic area is a reflection of its growing importance in discourse, and that the latter is in turn correlated with emergence of neologisms in that area. In terms of word embeddings, we again consider nearest word vectors as the word's semantic neighbors and quantify the rate at which their frequencies grow over decades (formally defined in §4.4).

4 Methodology

Our analysis is based on comparing embedding space neighborhoods of neologism word vectors and neighborhoods of embeddings of words from an alternative set. Automatic selection of neologisms is described in §4.2, and in §4.4 we detail the factors we control for when selecting the alternative set. In §4.1 we describe the datasets used in our experiments. Our data is split into two large corpora, HISTORICAL and MODERN; we additionally require the HISTORICAL corpus to be split into smaller time periods so that we can estimate word frequency change rate. Embedding models are trained on each of the two corpora, as described in §4.3. We compare the neighborhoods in the HISTORICAL embedding space, but due to the nature of our neologism selection process, many neologisms might not exist in the HISTORICAL vocabulary. To locate their neighborhoods, we adapt an approach from prior work in diachronic analysis with word embeddings: we learn an orthogonal projection between HISTORICAL and MODERN embeddings to align the two spaces in order to make them comparable (see Hamilton et al., 2016), and use projected vectors to represent neologisms in the HISTORICAL space. Finally, §4.5 describes the details of hypothesis testing: statistics we choose to quantify our two hypotheses and how their distributions are compared.

4.1 Datasets

We use the Corpus of Historical American English (COHA, Davies, 2002) and the Corpus of Contemporary American English (COCA, Davies, 2008), large diachronic corpora balanced by genre to reflect the variety in word usage. COHA data is split into decades; we group COHA documents from 18 decades (1800-1989) to represent the HISTORICAL English collection and use full COCA 1990-2012 corpus as MODERN.

The obtained HISTORICAL split contains 405M tokens of 2M types, and MODERN contains 547M tokens of 3M types.[2]

4.2 Neologism selection

We rely on a usage-based approach to extract the set of neologisms for our analysis, choosing the words based on their patterns of occurrence in our datasets. It can be seen as an approximation to selecting words based on their earliest recorded use dates, as these dates are also determined based on the words' usage in historical corpora. This analogy is supported by the qualitative analysis of the obtained set of neologisms, as discussed in §6.

We limit our analysis to nouns, an open-class lexical category. We identify nouns in our corpora using a part-of-speech dictionary, collected from a POS-tagged corpus of English Wikipedia data (Wikicorpus, Reese et al., 2010), and select words that are most frequently tagged as 'NN'.

We additionally filter candidate neologisms to exclude words that occur more frequently in capitalized than lowercased form; this heuristic helps us remove proper nouns missed by the POS tagger.

We select a set of neologisms by picking words that are substantially more frequent in the MODERN corpus than in the HISTORICAL one. It is important to note that while we use the term "neologism," implying a word at the early stages of emergence, with this method we select words that have entered mainstream vocabulary in MODERN time but might have been coined prior to that. We consider a word w to be a neologism if its ratio $f_m(w)/f_h(w)$ is greater than a certain threshold; here $f_m(\cdot)$ and $f_h(\cdot)$ denote word frequencies (normalized counts) in MODERN and HISTORICAL data respectively. Empirically we set the frequency ratio threshold equal to 20.

We rank words satisfying these criteria by their frequency in the MODERN corpus and select the first 1000 words to be our neologism set; this is to ensure that we only analyze words that subsequently become mainstream and not misspellings or other artifacts of the data.

4.3 Embeddings

Our hypothesis testing process involves inspecting semantic neighborhoods of neologisms in the HISTORICAL embedding space. However, many neologisms are very infrequent or nonexistent in the HISTORICAL data, so we approximate their vectors in the HISTORICAL space by projecting their MODERN embeddings into the same coordinate axes.

We learn Word2Vec Skip-Gram embeddings[3] (Mikolov et al., 2013) of the two corpora

[2] Statistics accompanying the corpora state that entire COHA dataset contains 385M words, and COCA contains 440M words; we assume the discrepancy is explained by tokenization differences.

[3] Hyperparameters: vector dimension 300, window size 5, minimum count 5.

and use orthogonal Procrustes to learn the aligning transformation:

$$\mathbf{R} = \arg\min_{\mathbf{\Omega}} \|\mathbf{\Omega}\mathbf{W}^{(m)} - \mathbf{W}^{(h)}\|,$$

where $\mathbf{W}^{(h)}, \mathbf{W}^{(m)} \in \mathbb{R}^{|V|\times d}$ are the word embedding matrices learned on the HISTORICAL and MODERN corpora respectively, restricted to the intersection of the vocabularies of the two corpora (i.e. every word embedding present in both spaces is used as an anchor). To project MODERN word embeddings into the HISTORICAL space, we multiply them by the obtained rotation matrix $\mathbf{R}$.

4.4 Control set selection

To test our hypotheses, we collect an alternative set of words and analyze how certain statistical properties of their neighbors differ from those of neighbors of neologisms. At this stage it is important to control for non-semantic confounding factors that might affect the word distribution in the semantic space. One such factor is word frequency: it has been shown that embeddings of words of similar frequency tend to be closer in the embedding space (Schnabel et al., 2015; Faruqui et al., 2016), which results in very dense clusters, or hubs, of words with high cosine similarity (Radovanović et al., 2010; Dinu et al., 2014). We choose to also restrict our control set to only include words that did not substantially grow or decline in frequency over the HISTORICAL period in order to prevent selecting counterparts that only share similar frequency in the MODERN subcorpus (e.g., due to recent topical relevance), but exhibit significant fluctuation prior to that period. In particular, we refrain from selecting words that emerged in language right before our HISTORICAL-MODERN split.

We create the alternative set by pairing each neologism with a non-neologism counterpart that exhibits a stable frequency pattern, while controlling for word frequency and word length in characters. Length is chosen as an easily accessible correlate to other factors for which one should control, such as morphological complexity, concreteness, and nativeness. We perform the pairing only to ensure that the distribution of those properties across the two sets is comparable, but once the selection process is complete we treat control words as a set rather than considering them in pairs with neologisms.

Following Stewart and Eisenstein (2018), we formalize frequency growth rate as the Spearman correlation coefficient between timesteps $\{1, \ldots, T\}$ and frequency series $f_{(1:T)}(w)$ of word w. In our setup, timesteps $\{1, \ldots, 18\}$ enumerate decades from 1810s to 1980s, and $f_t(\cdot)$ denote word frequencies in the corresponding t-th decade of the HISTORICAL data.

Formally, for each neologism w_n we select a counterpart w_c satisfying the following constraints:

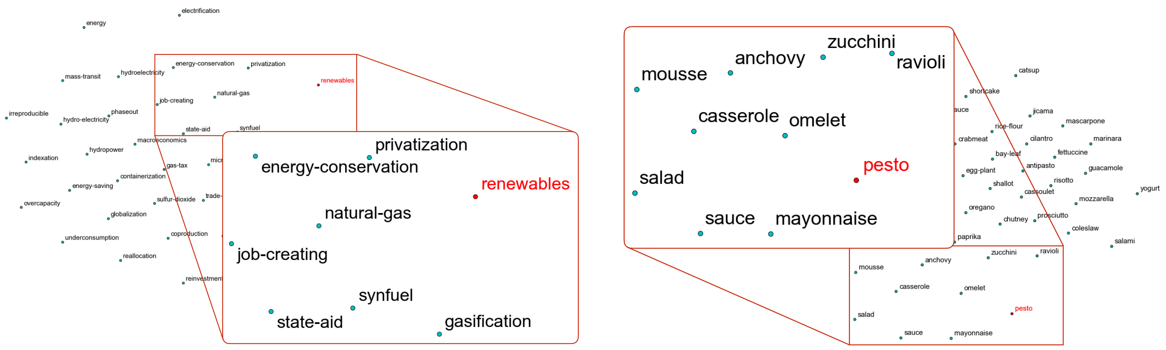
- Frequencies of the two words in the corresponding corpora are comparable: $f_m(w_n)/f_h(w_c) \in (1-\delta, 1+\delta)$, where δ was set to 0.25;
- The length of the two words is identical up to 2 characters;
- The Spearman correlation coefficient r_s between decades $\{1, \ldots, 18\}$ and the control word frequency series $f_{(1:18)}(w_c)$ is small: $|r_s\left(\{1:18\}, f_{(1:18)}(w_c)\right)| \leq 0.1$

These words, which we will refer to as *stable*, make up our default and most restricted control set. We will also compare neologisms to a *relaxed* control set, omitting the stability constraint on the frequency change rate but still controlling for length and overall frequency, to see how neologisms differ from non-neologisms in a broader perspective.

4.5 Experimental setup

We evaluate our hypotheses by inspecting neighborhoods of neologisms and their stable control counterparts in the HISTORICAL embedding space, viewing them as proxy for neighborhoods in the underlying semantic space. Since many neologisms are very infrequent or nonexistent in the HISTORICAL data, we approximate their vectors in the HISTORICAL space with their MODERN embeddings projected using the transformation described in §4.3. The neighborhood of a word w is defined as the set of HISTORICAL words for which cosine similarity between their HISTORICAL embeddings and v_w exceeds the given threshold τ; v_w denotes a projected MODERN embedding if w is a neologism or a HISTORICAL embedding if it is a control word.[4]

[4]Cosine similarity is chosen as our distance metric since it is traditionally used for word similarity tasks in distributional



(a) Semantic neighborhood of the word *renewables*.

(b) Semantic neighborhood of the word *pesto*.

Figure 1: Neighborhoods of projected MODERN embeddings of two neologisms (shown in red), *renewables* and *pesto*, in the HISTORICAL embedding space, visualized using t-SNE (Maaten and Hinton, 2008). Figure 1a shows an example of a neighborhood exhibiting frequency growth: words like *synfuel* or *privatization* have been used more towards the end of the HISTORICAL period. The neighborhood also includes *natural-gas* that can be seen as representing a concept to be replaced by *renewables*. The word *pesto* (Figure 1b) is projected into a neighborhood of other food-related words, most of which are also loanwords, several from the same language; it also has its hypernym *sauce* as one of its neighbors.

The two factors we need to formalize are semantic sparsity of the neighborhoods and increase of popularity of the topic that the neighborhood represents. We use sparsity in the embedding space as a proxy for semantic sparsity and approximate growth of interest in a topic with frequency growth of words belonging to it (i.e. embedded into the corresponding neighborhood). For the neighborhood of each word w, we compute the following statistics, corresponding to our two hypotheses:

1. *Density of a neighborhood* $d(w, \tau)$: number of words that fall into this neighborhood $d(w, \tau) = |\{u : \text{cosine}(v_w, v_u) \geq \tau\}|$
2. *Average frequency growth rate of a neighborhood* $r(w, \tau)$: as defined in the previous subsection, we compute the Spearman correlation coefficient between timesteps and frequency series for each word in the neighborhood and take their mean:

$$r(w, \tau) = \frac{1}{d(w, \tau)} \times \\ \times \sum_{u:\text{cosine}(v_w, v_u) \geq \tau} r_s\left(\{1:18\}, f_{(1:18)}(u)\right)$$

In our tests, we compare the values of those metrics for neighborhoods of neologisms and neighborhoods of control words and estimate the significance of each of the two factors for a range of neighborhood sizes defined by the threshold τ. We test whether means of the distributions of those statistics for the neologism and the control set differ and whether each of the two is significant for classifying words into neologisms and controls.

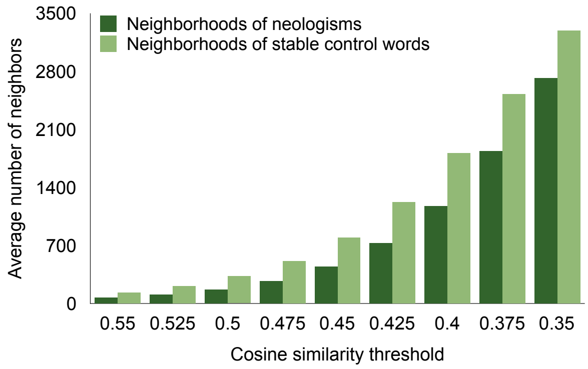
As mentioned in §4.2, our vocabulary is restricted to nouns, and we only consider vocabulary noun neighbors when evaluating the statistics.[5] Since we project all neologism word vectors from MODERN to HISTORICAL embedding space, for neologisms occurring in the HISTORICAL corpus we might find a HISTORICAL vector of the neologism itself among the neighbors of its projection; we exclude such neighbors from our analysis. We cap the number of nearest neighbors to consider at 5,000, to avoid estimating statistics on overly large sets of possibly less relevant neighbors.

5 Results

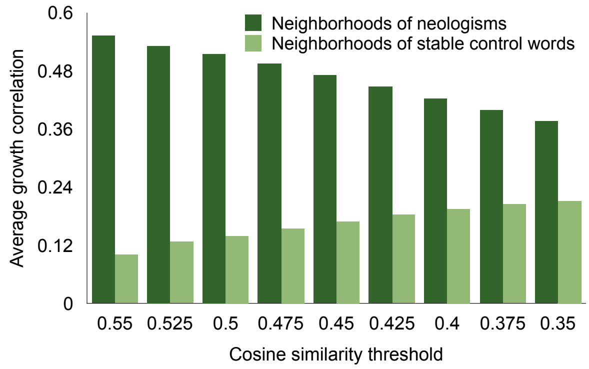
Following the experimental setup described in §4.5, we estimate the contribution of each of the hypothesized factors employing strictly constrained and relaxed control sets. We start by analyzing how the distributions of those statistics differ for neologisms and stable controls, both by

semantics (Lenci, 2018). We have also observed the same results when repeating the experiments with the Euclidean distance metric.

[5] Here we refer to the vocabulary of words participating in our analysis, not the embedding model vocabulary; embeddings are trained on the entire corpora.



(a) Average HISTORICAL word vector density in the neighborhoods of neologisms and stable control set words.



(b) Average frequency growth rate of HISTORICAL word vectors in the neighborhoods of neologisms and stable control set words.

Figure 2: Number of HISTORICAL word vectors within a certain cosine distance of a word and average growth rate of frequency (represented by Spearman correlation coefficient) of those HISTORICAL words, averaged across neologism (darker) and stable control word (lighter) sets. Projected neologism vectors appear in lower-density neighborhoods compared to control words, and neighbors of neologisms exhibit a stronger growth trend than those of the control words, especially in smaller neighborhoods.

comparing their sample means and by more rigorous statistical testing. We also evaluate the significance of the factors using generalized linear models for both stable and relaxed control sets.

5.1 Comparison to stable control set

First, we test our hypotheses on 720 neologism-stable control word pairs (not all words are paired in the stable control setting due to its restrictiveness).

Figure 2 demonstrates the values of density and frequency growth rate for a range of neighborhood sizes, averaged over neologism and control sets. Both results conform with our hypotheses: Figure 2a shows that on average the projected neologism has fewer neighbors than its stable counterpart, especially for larger neighborhoods, and Figure 2b shows that, on average, frequencies of neighbors of a projected neologism grow at a faster rate than those of a counterpart. Interestingly, we find that neighbors of stable controls still tend to exhibit small positive growth rate. We attribute it to the general pattern that we observed: about 70% of words in our vocabulary have positive frequency growth rate. We believe this might be explained by the imbalance in the amount of data between decades (e.g. 1980s sub-corpus has 20 times more tokens than 1810s): some words might not occur until later in the corpus because of the relative sparsity of data in the early decades.

As we can see from Figure 2a, neighborhoods of larger sizes (corresponding to lower values of the threshold) may contain thousands of words, so the statistics obtained from those neighborhoods might be less relevant; we might only want to consider the immediate neighborhoods, as those words are more likely to be semantically related to the central word. It is notable that the difference in the growth trends of the neighbors is substantially more prominent for smaller neighborhoods (Figure 2b): average correlation coefficient of immediate neighbors of stable words also falls into stable range as we defined it, while immediate neighbors of neologisms exhibit rapid growth.

5.2 Statistical significance

To estimate the significance and relative contribution of the two factors, we fit a generalized linear model (GLM) with logistic link function to the corresponding features of neologism and control word neighborhoods:[6]

$$y(w) \sim (1 + \exp(-\beta_0^{(\tau)} - \\ - \beta_d^{(\tau)} \cdot d(w,\tau) - \beta_r^{(\tau)} \cdot r(w,\tau)))^{-1}$$

where y is a Bernoulli variable indicating whether the word w belongs to the neologism set (1) or the control set (0), and τ is the cosine similarity threshold defining the neighborhood size.

Table 1 shows how the coefficients and p-values for the two statistics change with the neighborhood size. We found that when comparing with

[6]We use the implementation provided in the MATLAB Statistics and Machine Learning Toolbox.

Neighborhood size	Stable control set				Relaxed control set			
	Density		Growth		Density		Growth	
	$\beta_d^{(\tau)} \times 10^4$	p-value	$\beta_r^{(\tau)} \times 10$	p-value	$\beta_d^{(\tau)} \times 10^4$	p-value	$\beta_r^{(\tau)}$	p-value
Large ($\tau = 0.35$)	1.98	8.25×10^{-5}	1.84	2.35×10^{-80}	-1.07	5.63×10^{-4}	0.61	2.83×10^{-34}
Medium ($\tau = 0.45$)	0.20	8.29×10^{-1}	1.16	2.92×10^{-80}	-3.67	4.00×10^{-10}	0.46	6.19×10^{-46}
Small ($\tau = 0.55$)	6.90	2.90×10^{-2}	0.70	1.61×10^{-68}	-8.92	4.01×10^{-5}	0.28	1.19×10^{-36}

Table 1: Values of the GLM coefficients and their p-values for different neighborhood cosine similarity thresholds τ. $\beta_d^{(\tau)}$ and $\beta_r^{(\tau)}$ denote the coefficients for density and average frequency growth respectively for neighborhoods defined by τ. Comparing the results for the stable and relaxed control sets, we find that for the stable controls density is only significant in larger neighborhoods, but without the stability constraint both factors are significant for all neighborhood sizes.

the stable control set, average frequency growth rate of the neighborhood was significant for all sizes, but neighborhood density was significant at level $p < 0.01$ only for the largest ones.[7] We attribute this to the effect discussed in the previous section: difference in average frequency growth rate between neighbors of neologisms and stable words shrinks as we include more remote neighbors (Figure 2b), so for large neighborhoods frequency growth rate by itself is no longer predictive enough.

We also evaluate the significance of features for the relaxed control set without the stability constraint on 1000 neologism-control pairs. We have repeated the experiment with 5 different randomly sampled relaxed control sets (results for one showed in Table 1). For medium-sized neighborhoods ($0.4 \leq \tau \leq 0.5$) density variable is always significant at $p < 0.01$, but densities of largest and smallest neighborhoods were rejected in several runs. With more variance in the control set, differences in neighborhood frequency growth rate between neologisms and controls are less prominent than in the stable setting, so density plays a more important role in prediction.[8]

Growth feature weights $\beta_r^{(\tau)}$ are always positive and density feature weights $\beta_d^{(\tau)}$ are negative in the relaxed setting (where density is significant). This matches our intuition that neighborhood frequency growth and sparsity are predictive of neology.

Comparing sample means of density and growth rates between neologisms and each of the 5 randomly selected relaxed control sets (as we did for stable controls in Figure 2) demonstrated that neologisms still appear in sparser neighborhoods than the controlled counterparts. The difference in frequency growth rate between the neologism and control word neighborhoods is also observed for all control sets (although it varies noticeably between sets), but it no longer exhibits an inverse correlation with neighborhood size.

6 Discussion

We have demonstrated that our two hypotheses hold for the set of words we automatically selected to represent neologisms. To establish validity of our results, we qualitatively examine the obtained word list to see if the words are in fact recent additions to the language. We randomly sample 100 words out of the 1000 selected neologisms and look up their earliest recorded use in the Oxford English Dictionary Online (OED, 2018). Of those 100 words, eight are not defined in the dictionary: they only appear in quotations in other entries (*bycatch* (quotation from 1995), *twentysomething* (1997), *cross-sex* (1958), etc.) or do not occur at all (*all-mountain*, *interobserver*, *off-task*). Of the remaining 92 words, 78 have been first recorded after the year 1810 (i.e. since the beginning of the HISTORICAL timeframe), 44 have been first recorded in the twentieth century, and 21 words since 1950. However, some of the words dating back to before 19th century have only been recorded in their earlier, possibly obsolete sense: for example, while there is evidence of the word *software* being used in 18th century, this usage corresponds to its obsolete meaning of 'textiles, fabrics', while the first recorded use in its currently dominant sense of 'programs essential to the operation of a computer system' is dated 1958. To account for such semantic neologisms, we can count

[7] Applying Wilcoxon signed-rank test to the series of neighborhood density and frequency growth values for neologism and stable control sets showed the same results.

[8] Detailed results of the regression analysis and collinearity tests can be found in the repository. No evidence of collinearity was found in any of the experiments.

the first recorded use of the newest sense of the word; that gives us 82, 58 and 31 words appearing since 1810, 1900 and 1950 respectively.[9] This leads us to assume that most words selected for our analysis have indeed been neologisms sometime over the course of the HISTORICAL time.

We would also like to note that the results of this examination may be skewed due to factors for which lexicography may not account: for example, many words identified as neologisms are compound nouns like *countertop* or *soundtrack* that have been written as two separate words or joined with a hyphen in earlier use. There is also considerable spelling variation in loanwords, e.g. *cuscusu*, *cooscoosoos*, *kesksoo* were used interchangeably before the form *couscous* was accepted as the standard spelling. Specific word forms might also have different life cycles: while the word *music* existed in Middle English, the plural form *musics* in a particular sense of 'genres, styles of music' is much more recent.

Qualitative examination of the neologism set reveals that new words tend to appear in the same topics; for example, many words in our set were related to food, technology, or medicine. This indirectly supports our second hypothesis: rapid change in these spheres makes it likely for related terms to substantially grow in frequency over a short period of time. One example of such a neighborhood is shown in Figure 1a: the neologism *renewables* appeared in a cluster of words related to energy sources — a topic that has been more discussed recently. There is also some correlation between the topic and how new words are formed in it: most food neologisms are so-called cultural borrowings (Weinreich, 2010), when the name gets loaned from another culture together with the concept itself (e.g. *pesto*, *salsa*, *masala*), while many technology neologisms are compounds of existing English morphemes (e.g. *cyber+space*, *cell+phone*, *data+base*).

We also consider nearest neighbors (HISTORICAL words with highest cosine similarity) of the neologisms to ensure that they are projected into the appropriate parts of the embedding space. Examples of nearest neighbors are shown in Table 2. We saw different patterns of how the concept represented by the neologism relates to concepts represented by its neighbors. For example, some terms for new concepts appear next to related concepts they succeeded and possibly made obsolete: e.g. *email:letter*, *e-book:paperback*, *database:card-index*. Other neologisms emerge in clusters of related concepts they still equally coexist with: *hip-hop:jazz*, *hoodie:turtleneck*; most cultural borrowings fall under this type (see the neighborhood of *pesto* in Figure 1b). Both those patterns can be viewed as examples of a more general trend: one concept takes place of another related one, whether in terms of fully replacing it or just taking its place as the dominant form.

Neologism	Nearest HISTORICAL neighbors	
email	telegram	letter
pager	beeper	phone
blogger	journalist	columnist
sitcom	comedy	movie
spokeswoman	spokesman	director
sushi	caviar	risotto
rehab	detoxification	aftercare

Table 2: Nearest HISTORICAL neighbors of projected MODERN embeddings for a sample of emerging words. We can see that words get projected into semantically relevant neighborhoods, and nearest neighbors can even be useful for observing the evolution of a concept (e.g. *pager:beeper*).

Other interesting effects we observed include lexical replacement (a new word form replacing an old one without a change in meaning, e.g. *vibe:ambience*), tendency to abbreviate terms as they become mainstream (*biotech:biotechnology*, *chemo:chemotherapy*), and the previously mentioned changes in spellings of compounds (*lifestyle:life-style*, *daycare:day-care*).

7 Conclusion

We have shown that our two hypothesized factors, semantic neighborhood sparsity and its average frequency growth rate, play a role in determining in what semantic neighborhoods new words are likely to emerge. Our analyses provide more support for the latter, conforming with prior linguistic intuition of how language-external factors (which this factor implicitly represents) affect language change. We also found evidence for the former, although it was found less significant.

Our contributions are manifold. From a computational perspective, we extend prior research

[9]For all words that have one or more senses marked as a noun, we only consider those senses. Out of the 92 listed words, only three do not have nominal senses, and for two more usage as a noun is marked to be rare.

on meaning change to a new task of analyzing word emergence, proposing another way to obtain linguistic insights from distributional semantics. From the point of view of linguistics, we approach an important question of whether language change is affected by not only language-external factors but language-internal factors as well. We show that internal factors—semantic sparsity, specifically—contribute to where in semantic space neologisms emerge. To the best of our knowledge, our work is the first to use word embeddings as a way of quantifying semantic sparsity. We have also been able to operationalize one kind of external factor, technological and cultural change, as something that can been measured in corpora and word embeddings, paving the way to similar work with other kinds of language-external factors in language change.

An admittable limitation of our analysis lies in its restricted ability to account for polysemy, which is a pervasive issue in distributional semantics studies (Faruqui et al., 2016). As such, semantic neologisms (existing words taking on a novel sense) were not a subject of this study, but they introduce a potential future direction. Additional properties of word's neighbors can also be correlated with word emergence, both language-internal (word abstractness or specificity) and external; these can also be promising directions for future work. Finally, our future plans include exploration of how features of semantic neighborhoods are correlated with word obsolescence (gradual decline in usage), using similar semantic observations.

Acknowledgments

We thank the BergLab members for helpful discussion, and the anonymous reviewers for their valuable feedback. This work was supported in part by NSF grant IIS-1812327.

References

Jean Aitchison. 2001. *Language Change: Progress Or Decay?* Cambridge University Press.

René Appel and Pieter Muysken. 2006. *Language contact and bilingualism*. Amsterdam University Press.

Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *International Conference on Machine Learning*, pages 380–389.

Michel Bréal. 1904. *Essai de sémantique:(science des significations)*. Hachette.

Lyle Campbell. 2013. *Historical Linguistics: an Introduction*. MIT Press, Cambridge, MA.

Paul Cook. 2012. Using social media to find English lexical blends. In *Proceedings of the 15th EURALEX International Congress (EURALEX 2012)*, pages 846–854, Oslo, Norway.

Frank E. Daulton. 2012. Lexical borrowing. In *The Encyclopedia of Applied Linguistics*. American Cancer Society.

Mark Davies. 2002. *The Corpus of Historical American English (COHA): 400 million words, 1810-2009*. Brigham Young University.

Mark Davies. 2008. *The corpus of contemporary American English*. BYE, Brigham Young University.

Marco Del Tredici and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1591–1603.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2017. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1136–1145.

Jacob Eisenstein. 2017. Identifying regional dialects in on-line social media. *The Handbook of Dialectology*, pages 368–383.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2012. Mapping the geographical diffusion of new words. In *NIPS Workshop on Social Network and Social Media Analysis*.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1489–1501.

Kokil Jaidka, Niyati Chhaya, and Lyle Ungar. 2018. Diachronic degradation of language models: Insights from social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 195–200.

Daniel Kershaw, Matthew Rowe, and Patrick Stacey. 2016. Towards modelling language innovation acceptance in online social networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 553–562. ACM.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171.

Rochelle Lieber. 2017. Derivational morphology.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Michael Proffitt, editor. 2018. *OED Online*. Oxford University Press. http://www.oed.com/.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Samuel Reese, Gemma Boleda, Montse Cuadros, Lluís Padró, and German Rigau. 2010. Wikicorpus: A word-sense disambiguated multilingual Wikipedia corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Vincent Renner, Franois Maniez, and Pierre Arnaud, editors. 2012. *Cross-disciplinary perspectives on lexical blending*. De Gruyter Mouton, Berlin.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Ian Stewart and Jacob Eisenstein. 2018. Making" fetch" happen: The influence of social and linguistic context on nonstandard word growth and decline. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4370.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to diachronic conceptual change. *arXiv preprint arXiv:1811.06278*.

John R Taylor and Anthony P. Grant. 2014. *Lexical Borrowing*. Oxford University Press, Oxford.

Uriel Weinreich. 2010. *Languages in contact: Findings and problems*. Walter de Gruyter, The Hague.

Yang Xu and Charles Kemp. 2015. A computational evaluation of two laws of semantic change. In *CogSci*.

MG Parsing as a Model of Gradient Acceptability in Syntactic Islands

Aniello De Santo
Department of Linguistics
Institute for Advanced Computational Science
Stony Brook University
aniello.desanto@stonybrook.edu

Abstract

It is well-known that the acceptability judgments at the core of current syntactic theories are continuous. However, an open debate is whether the source of such gradience is situated in the grammar itself, or can be derived from extra-grammatical factors. In this paper, we propose the use of a top-down parser for Minimalist grammars (Stabler, 2013; Kobele et al., 2013; Graf et al., 2017), as a formal model of how gradient acceptability can arise from categorical grammars. As a test case, we target the acceptability judgments for island effects collected by Sprouse et al. (2012a).

1 Introduction

The human judgments linguists use to evaluate the adequacy of syntactic theories fall in a wide, non-binary spectrum of acceptability — a fact well-known from the early days of generative grammar (Chomsky, 1956, 1965, a.o.). Nonetheless, mainstream syntax has long claimed that grammatical knowledge is, at its core, categorical, and that *gradience* in acceptability judgments comes from extra-grammatical factors (Sprouse, 2007, a.o.). However, the rise of experimental methods in theoretical syntax has renewed the question of whether gradience should be integrated in grammatical theories directly, for instance in the form of probabilistic models (Keller, 2000; Crocker and Keller, 2005; Sorace and Keller, 2005; Lau et al., 2014, 2015, 2017).

As the relation between grammaticality and acceptability is not transparent, constructing a well-specified theory of how gradient acceptability arises from grammatical knowledge is clearly valuable. From an empirical perspective, however, categorical approaches seem to be at a disadvantage when compared to gradient grammatical models rooted in quantitative, probabilistic frameworks.

There is an abundance of well-known proposals about the way syntactic structure and cognitive resources can be integrated to derive connections between acceptability and processing difficulty (e.g., Yngve, 1960; Wanner and Maratsos, 1978; Rizzi, 1990; Rambow and Joshi, 2015; Gibson, 2000; McElree et al., 2003; Lewis and Vasishth, 2005, a.o.). However, few models based on current grammatical formalisms have been implemented in precise computational frameworks (cf. Boston, 2010). In order to have a complete theory of how acceptability judgments correlate to categorical grammars, what seems to be necessary is a formal model of the syntactic structures licensed by said grammars, and a theory of how such structures interact with extra-grammatical factors to derive differences in acceptability. This would make it possible to test how assumptions about fine-grained syntactic details lead to quantifiable predictions for the gradient acceptability of individual sentences (Stabler, 2013; Sprouse et al., 2018).

Here, we suggest that a parser for Minimalist grammars (MGs; Stabler, 2013), coupled with complexity metrics measuring memory usage (Kobele et al., 2013; Graf et al., 2017, a.o.), is an effective model to address these issues. The MG parser has been used in the past to study which aspects of grammar drive processing cost for a vast set of off-line processing asymmetries cross-linguistically (Gerth, 2015; Graf et al., 2017; Zhang, 2017). Given the ability of MGs to encode rich syntactic analyses, the MG parser is especially sensitive to fine-grained grammatical information, and thus is able to generate quantitative predictions especially suited to our purposes.

In particular, we relate sentence acceptability to sentence structure by specifying: 1) a formalized theory of syntax in the form of MGs; 2) a parser as a model of how the structural representation of a sentence is built from its linear form; 3) a linking theory between structural complexity and acceptability in the form of metrics measuring memory usage. As a proof-of-concept for the validity of the linking theory, we model the acceptability judgments for three types of syntactic islands, using as a baseline the judgments reported in (Sprouse et al., 2012a).

Importantly, our main aim is not to settle the debate of whether gradience should be found in the grammar itself, or in the interaction between grammar and external factors (if such a debate could ever be settled). What we offer is a formalized, testable model of the latter hypothesis, in the hope of providing ground for a more principled investigation of categorical grammaticality and continuous acceptability.

2 MG Parsing

2.1 MGs

MGs (Stabler, 1997, 2011) are a lexicalized, mildly context-sensitive formalism incorporating the structurally rich analyses of Minimalist syntax — the most recent version of Chomsky's transformational grammar.

An MG grammar is a set of lexical items (LIs) consisting of a phonetic form and a finite, non-empty string of features. LIs are assembled via two feature checking operations: *Merge* and *Move*. Intuitively, Merge encodes subcategorization, while Move encodes long-distance movement dependencies. Here, we avoid most of the technical details of the formalism, and we limit our discussion to a general description of the data structures defined by these grammars.

MGs' *derivation trees* encode the sequence of Merge and Move operations required to build the phrase structure tree for a specific sentence (Michaelis, 1998; Harkema, 2001). In a traditional derivation tree, all leaf nodes are labeled by LIs, while unary and binary branching nodes are labeled as Move or Merge, respectively. However, as the details of the feature calculus are irrelevant to us, we adopt a simpler representation that discards the feature annotation of LIs, and labels internal nodes as standard in minimalist syntax. We also explicitly include dashed arrows indicating movement relations.[1]

[1]Note that, due to the fact that intermediate landing sites for moved phrases do not affect the traversal strategy, we do not explicitly highlight them with movement arrows.

The fundamental difference between a phrase structure tree and a derivation tree is that in the latter, moved phrases remain in their base position, and their landing site must be fully reconstructed via the feature calculus (cf. Fig. 1a and Fig. 1b). As a concequence, the final word order of a sentence is not directly reflected in the order of the leaf nodes in a derivation tree.

Importantly, MG derivation trees form a regular tree language, and thus can be regarded as a simple variant of context-free grammars (CFG), allowing us to exploit some of CFGs more established parsing algorithms.

2.2 Top-down MG Parsing

We follow recent sentence processing results, and adopt Stabler (2013)'s top-down parser for MGs. This parser is a variant of a standard depth-first, top-down parser for CFGs: it takes as input the string representation of a sentence, hypothesizes the structure top-down, verifies that the words in the structure match the input string, and outputs an encoding of the sentence structure in the form of a derivation tree. Importantly, the surface order of lexical items in the derivation tree is not the phrase structure tree's surface order. Thus, simple top-to-bottom and left-to-right scanning of the leaf nodes yields the wrong word order. While scanning the nodes then, the MG parser must also keep track of the derivational operations which affect the linear word order.

Memory plays a crucial role in this procedure: if a node is hypothesized at step i, but cannot be worked on until step j, it must be stored for $j-i$ steps in a priority queue. To make this traversal strategy transparent to the reader, we adopt Kobele et al. (2013)'s notation, in which each node in the tree is annotated with an *index* (superscript) and an *outdex* (subscript). Intuitively, the annotation indicates for each node in the tree when it is first conjectured by the parser (index) and placed in the memory queue, and at what point it is considered completed and flushed from memory (outdex). Consider the tree in Fig. 1b, explicitly annotated with the parsing steps. The node *does* is hypothesized at step 3. However, *which engineer* comes before it in the input, so *does* has to wait until step 12 to be flushed out of the queue.

Finally, note that Stabler's parser was originally given a search beam discarding the most unlikely predictions. Here though, we are not interested

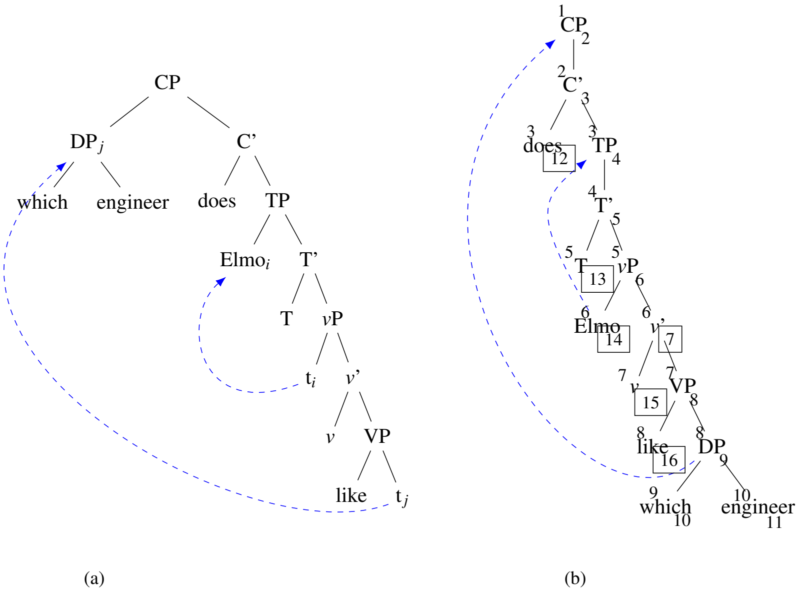


Figure 1: Phrase structure tree (a), and annotated MG derivation tree (b) for *Which engineer does Elmo like?*. Boxed nodes in (b) are those with tenure value greater than 2, following (Graf and Marcinek, 2014).

in the cost of choosing among alternative parsing choices, and want to focus on the specific contribution of the grammar to memory usage. Thus, we assume that the parser is equipped with a perfect oracle, which always makes the right choices when constructing a tree (Kobele et al., 2013). Essentially, the MG model employs a deterministic parsing strategy, where ambiguity has no role.

2.3 Measuring Memory Usage

Recently, Stabler (2013)'s MG parser has been used to investigate which aspect of grammatical structure affect off-line processing difficulty (Kobele et al., 2013; Graf and Marcinek, 2014; Gerth, 2015; Graf et al., 2017, a.o.).

In order to allow for psycholinguistic predictions, the behavior of the parser is related to processing difficulty via complexity metrics measuring how the structure of a tree affects memory. The MG model refers to three main notions of memory usage (Graf et al., 2017): (a) how long a node is kept in memory (*tenure*); (b) how many nodes must be kept in memory (*payload*); (c) how much information is stored in a node (*size*).

Tenure and payload for each node n in the tree can be easily computed via the node annotation scheme of Kobele *et al.*: a node's tenure is equal to the difference between its index and its outdex; the payload of a derivation tree is computed as the number of nodes with a tenure strictly greater than 2 (boxed nodes in our tree annotation scheme).[2] For instance, tenure for the node *does* in Fig. 1b is computed as $12-3=9$.

Defining size in an informal way is slightly trickier, as it was originally based on how information about movers is stored by Stabler's top-down parser (for a technical discussion, see Graf et al., 2015). In practice, size measures the hierarchical length of a movement dependency, and is computed as the index of a mover minus the index of its target site. Considering again the tree in Fig. 1b, the size of *Elmo* is $6-3=3$.

In order to contrast derivations, past work has used these general concepts to define a vast set of complexity metrics measuring processing difficulty over a full tree (Kobele et al., 2013). For instance, tenure can be associated to metrics like MAXT $:=$ $max(\{tenure\text{-}of(n)\})$ and SUMT $:=$ $\sum_n$ $tenure\text{-}of(n)$. MAXT measures the maximum amount of time any node stays in memory during processing, while SUMT measures the overall amount of memory usage for all nodes whose tenure is not trivial. It thus captures total memory usage over the course of a parse. As an illustrative example, consider one last time the tree in Fig. 1b. Tenure in this tree is mostly driven by the movement of the embedded object, thus MAXT is mea-

[2] We refer to tenure values ≤ 2 as *trivial*, since it arises naturally from the binary nature of derivation trees, and it's not due to extra waiting time in the priority queue (Graf and Marcinek, 2014).

sured at *does* and it is equal to $12-3=9$. Similar metrics can be defined for size. For instance, in Fig. 1b SUMS is given by the length of the object movement and the length of the subject movement: $(8-1)+(6-3)=10$.

These metrics have been surprisingly successful in accounting for a vast array of different processing phenomena, such as right embedding vs. center embedding, nested dependencies vs. crossing dependencies, as well as a set of contrasts involving relative clauses (Graf and Marcinek, 2014; Graf et al., 2015). However, Graf et al. (2015) argue that a better approach would make use of ranked metrics of the type $\langle M_1, M_2, \ldots, M_n \rangle$. Such rankings work in a way similar to constraint ranking in Optimality Theory (Prince and Smolensky, 2008): a lower ranked metric matters only if all higher ranked metric have failed to pick out a unique winner (e.g., if two constructions result in a *tie* over MAXT). Following this idea, Graf et al. (2017) show that when complexity metrics are allowed to be ranked in such a way the space of possible metrics quickly explodes (up to 1600 distinct metrics). Considering the total number of possible metrics, it is conceivable that some metric combination could explain any hypothetical processing asymmetry — thus reducing the explanatory power of the model. However, this does not seem to be the case. Graf et al. (2017) rule out the vast majority of these metrics, by showing their insufficiency in accounting for some crucial constructions across a variety of grammatical analyses.

Here then, we rely on previous work and focus on the predictions made by a ranked version of ⟨MAXT, SUMS⟩ in comparing memory burden for contrasting sentences (Zhang, 2017; Liu, 2018; Lee, 2018; De Santo, 2019; De Santo and Shafiei, 2019). In addition, our core linking hypothesis connects processing difficulty to acceptability by assuming that higher memory cost implies lower acceptability.

3 Gradient Acceptability in Syntactic Islands

Given the metrics' sensitivity to minor differences in syntactic structure, the MG parser's predictions are the most interpretable when used to compare the relative complexity of minimally different sentences. Careful comparisons across sentences as similar as possible in their underlying syntactic structure seem also to be desirable if we want to understand the source of gradient variation in acceptability judgments. For these reasons, we chose to model the data on the acceptability of syntactic islands collected by Sprouse et al. (2012a) (henceforth SWP), in a first investigation of the viability of the parser as a model of gradient acceptability.

Syntactic islands are well-known in linguistics (Chomsky, 1965; Ross, 1968) as a set of phenomena in which the acceptability of a sentence is degraded, in relation to the interaction of a long-distance dependency and its syntactic context. Consider the following sentences:

(1) a. What$_i$ did John say Bill saw t_i ?
    b. What$_i$ did John have dinner before Bill saw t_i?

In 1a, *what* is displaced from its lower position as the object of the verb *saw* to a sentence initial position. In 1b, this same displacement cannot take place, as *what* is inside an adjunct clause (headed by *because*). Thus, 1b is considered ill-formed by native speakers of standard American English. Since displacing an element from inside an adjunct leads to ungrammaticality, adjunct clauses are classic example of island structures.

SWP conducted an extensive investigation of the acceptability of island constructions, by collecting formal acceptability judgments for four island types using a magnitude estimation task. The acceptability contrasts in this study are optimal for our purposes for multiple reasons. First, while a categorical grammar would predict a binary split in sentence acceptability (violates an island/doesn't violate an island), the continuous scale the estimation task was based upon revealed a spectrum of gradient judgments. Second, the stimuli in SWP's design were based on a (2×2) factorial definition of island effects, and explicitly identify two structural factors that might affect acceptability: 1) the length of a movement dependency; 2) the presence of a so-called "island construction" (Kluender and Kutas, 1993). This careful dimensional decomposition of the test sentences, coupled with the continuous scale of the judgment task, resulted in a set of well-defined pairwise comparisons ideal for the MG parser's modeling approach.

In what follows, we test whether the gradient of acceptability shown in SWP's data is predicted by a parser grounded in a rich categorical grammar. Before proceeding with our analysis though, it seems to be important to make an additional note about our aims. An expert reader might know that there is an ongoing debate in the literature about the nature of islands effects (see, for instance, Hofmeister et al., 2012a; Sprouse et al., 2012b; Hofmeister et al., 2012b, and references therein) — with classical syntactic accounts rooting them in grammatical constraints, while others arguing that such effects can be reduced to a conspiracy of processing factors.

Importantly, we are *not* attempting to reduce these effects to processing demands and, at least at this stage, it is not our purpose to directly engage with this debate. For the same reasons, we do not investigate the *super-additivity* found in SWP's paper, as we are *not* interested in modeling the grammaticality of an island violation per-se. Relatedly, we do not claim that the acceptability of island violations is *purely* syntactic in nature, as it has been shown to be sensitive to a variety of semantic factors (Truswell, 2011; Kush et al., 2018; Kohrt et al., 2018, a.o.). Crucially, we are "just" interested in exploring the idea that the *gradient* component of acceptability judgments arises due to processing factors. We focus on islands effects exclusively because of the optimal baseline offered by SWP's data.

We will return to the question of whether our model could give *any* insights into the question of separating processing and grammatical contributions to island effects in Sec. 5.

4 Modeling Results

SWP focused on English wh-movement dependencies to explore four types of islands constructions: Subject, Adjunct, Complex NP, and Whether islands. Since the MG parser is only sensitive to structural differences, in this paper we ignore the case of Whether islands and concentrate on the remaining three cases. Table 1 presents a summary of all modeling contrasts in the paper, compared with the experimental results of SWP.[3]

4.1 Subject Island: Case 1

First, we model Subject islands as in SWP's Experiment 1, comparing 4 sentence types across 2 conditions: subject/object extraction, and island/non-island. Note that here *island* does not imply a violation, but refers to the presence of an island structure (Kluender and Kutas, 1993).

[3]All scripts are available at https://github.com/CompLab-StonyBrook/mgproc.

Island Type	Sprouse et al. (2012)	MG Parser
Subject Island Case 1	2b > 2a	✓
	2b > 2d	✓
	2b > 2c	✓
	2a > 2c	✓
	2a > 2d	✓
	2c > 2d	2c < 2d
Subject Island Case 2	3a > 3b	✓
	3a > 3c	✓
	3a > 3d	✓
	3b > 3d	✓
	3c > 3b	✓
	3c > 3d	✓
Adjunct Island	4a > 4b	✓
	4a > 4c	✓
	4a > 4d	✓
	4b > 4d	✓
	4c > 4b	✓
	4c > 4d	✓
Complex NP Island	5a > 5b	✓
	5a = 5c	✓
	5a > 5d	✓
	5b > 5d	✓
	5c > 5b	✓
	5c > 5d	✓

Table 1: Summary of results (as pairwise comparisons) from (Sprouse et al., 2012a), and corresponding parser's predictions (x > y: x more acceptable than y).

(2) a. What do you think the speech interrupted *t*? **Obj/Non Island**

b. What do you think *t* interrupted the show? **Subj/Non Island**

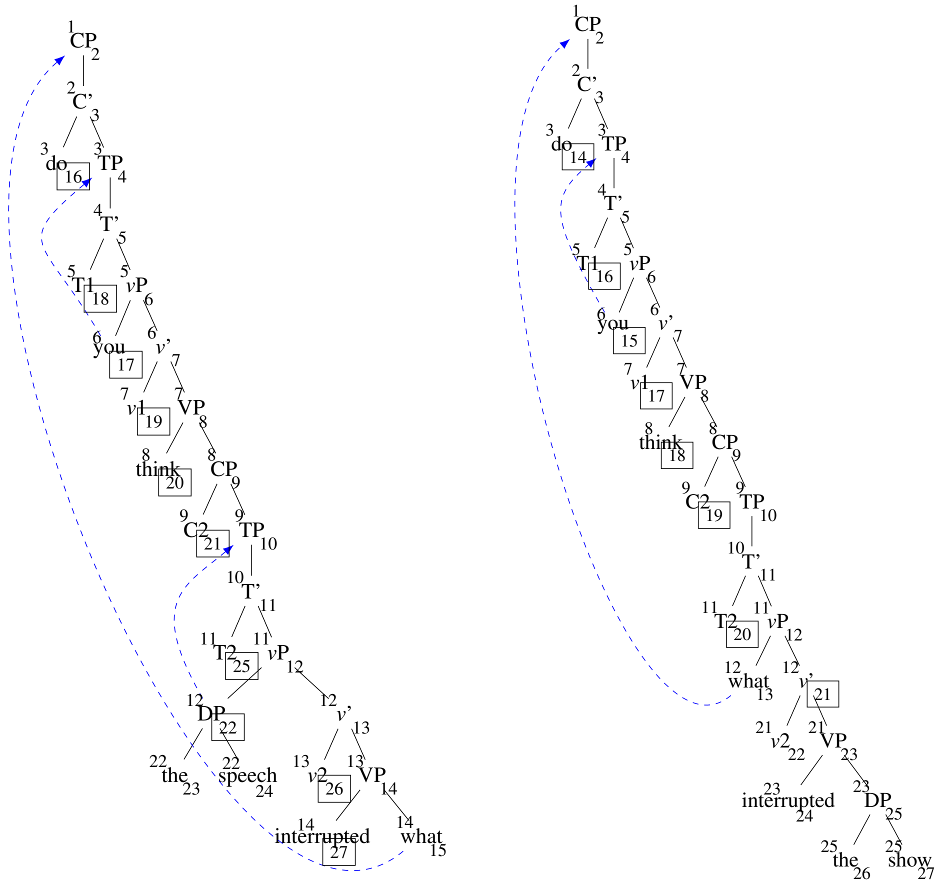
c. What do you think the speech about global warming interrupted the show about *t*? **Obj/Island**

d. What do you think the speech about *t* interrupted the show about global warming? **Subj/Island**

Annotated MG derivation trees for these sentences are shown in Fig. 2 (object/subject with no island) and Fig. 3 (with island).[4] The parser's predictions (via MAXT) overall match the experimental results (see Table 1).[5]

[4]Due to space constraints, annotated derivations are provided just for the Subject island case, as an illustrative example. Derivations for all other island types can be easily reconstructed from standard minimalist analyses of the test sentences (e.g., Adger, 2003). Source files can also be found at https://github.com/aniellodesanto/mgproc/tree/master/islands.

[5]When a wh-element is displaced from an embedded position, we avoid intermediate landing sites due to successive cyclicity. As intermediate movement steps do not affect the



(a) (b)

Figure 2: Annotated derivation trees for (a) 2a (object, non-island) and (b) 2b (subject, non-island).

The factorial design of the original study helps us understand the model's predictions. The contrast between 2b and 2a,2d is correctly captured by MAXT. This is due to the wh-element spanning a longer, more complex structure comprising the whole embedded DP subject in the Island cases. Compare 2a and 2b, both with highest tenure on *do* (14 and 11, respectively — cf. Tbl. 2). In 2a, *do* is conjectured after *what* has been scanned from the input. But then it cannot be flushed out of memory until *what* is confirmed in its base position as the embedded complement. In 2b, *do* only has to wait until the embedded subject position is reached, and then it is discarded from memory.

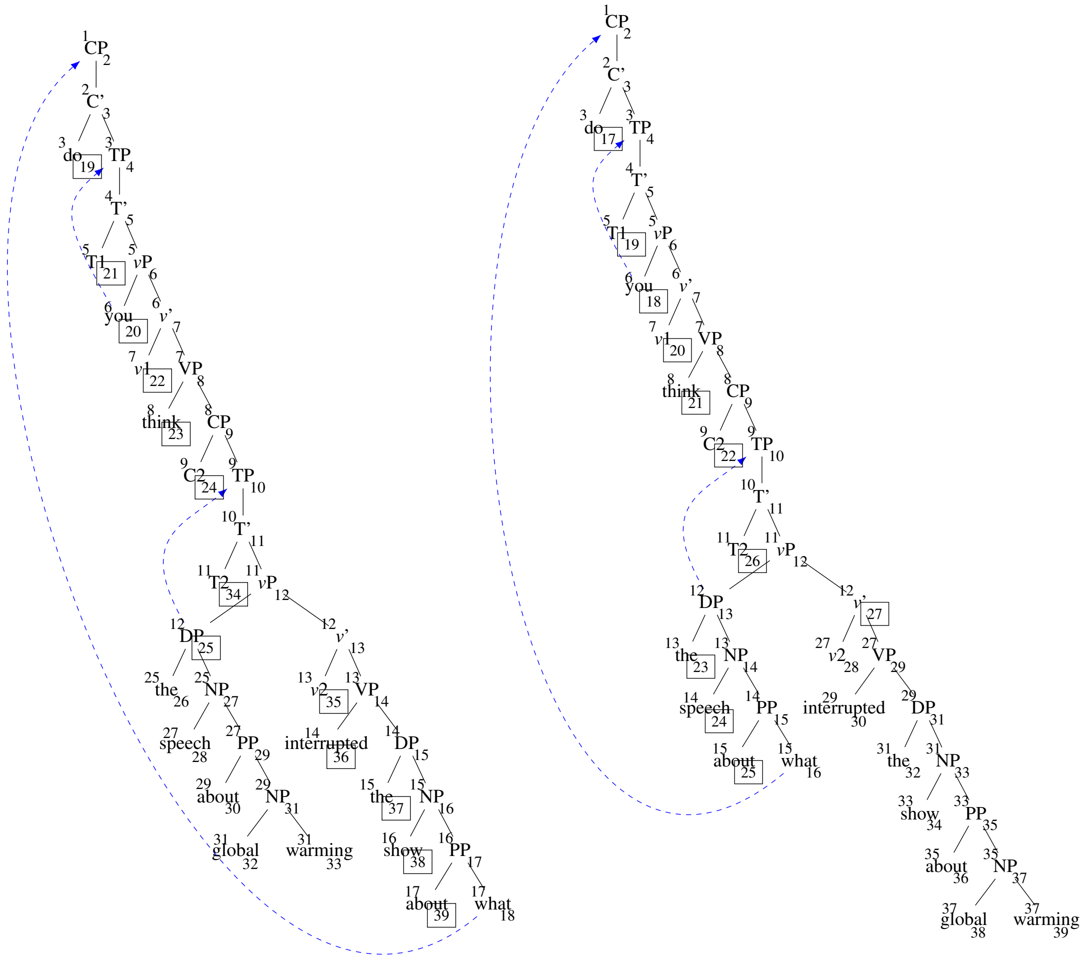
Consider now 2c. Here the highest tenure is on the embedded *T* head, which has to wait for the wh-element in object position, and then for the whole complex DP in subject position, before it can finally be flushed out of the queue. The longer wh-dependency in the object case explains once again why 2b is preferred over 2c, and the additional complexity of the DP subject is crucial in driving the 2b > 2c contrast.

Clause Type	Ex. #	MaxT	SumS
Obj./Non Island	2a	14/*do*	19
Subj./Non Island	2b	11/*do*	14
Obj./Island	2c	23/*T2*	22
Subj./Island	2d	15/*do*	20
Short/Non Island	3a	5/*C*	9
Long/Non Island	3b	11/*do*	14
Short/ Island	3c	11/*T2*	9
Long/ Island	3d	17/*T2*	20

Table 2: Summary of MAXT (*value/node*) and SUMS by test sentence for Subject island in case 1 and 2 (*T2* marks the embedded *T* head.)

Finally, there is one case in which parser's predictions and experimental data disagree: the contrast between subject and object extraction in the island condition (2c vs 2d). The parser predicts that 2c should be more acceptable than 2d (*Subj/Island > Obj/Island*). This is not surprising,

traversal strategy, this choice does not significantly change our results (cf. Zhang, 2017).



(a) (b)

Figure 3: Annotated derivation trees for the test sentences in (a) 2c (object, island) and (b) 2d (subject, island).

as the memory metrics pick up on the additional length of the extraction in the object case, and thus obviously predict the preference for a subject gap. However, SWP show *Obj/Island* > *Subj/Island* — which is expected from a theoretical perspective since 2d is the ungrammatical condition (i.e., there is an extraction out of an island).

We will come back to the significance of this mismatch in Sec. 5. Crucially for our main claim though, the parser correctly predicts the gradient of acceptability for those conditions that, according to a categorical grammar, should all be equivalent (i.e., those containing no forbidden extraction).

4.2 Subject Island: Case 2

The previous section suggests that, when a grammatical violation coincides with processing factors (e.g., length of a dependency), parser and human judgments should match on all contrasts. Luckily, SWP offer us the chance to test such a prediction, with a second set of subject island sentences. SWP's Experiment 2 compares a *short* dependency and *long* dependency (*matrix* vs *embedded* extraction in the original paper), again in an island and non-island condition.

(3) a. Who *t* thinks the speech interrupted the primetime TV show? **Short/Non Island**

b. What do you think *t* interrupted the primetime TV show? **Long/Non Island**

c. Who *t* thinks the speech about global warming interrupted the primetime TV show? **Short/Island**

d. What do you think the speech about *t* interrupted the primetime TV show? **Long/Island**

As expected, parser's preferences and experimental data fully match in this case, as the ungrammatical condition (3d) is also the one in which the movement dependency is the longest. Here however, deriving the correct preferences requires the ranking of ⟨MaxT,SumS⟩, instead of just MaxT alone (note also that SumS by itself would not suffice, as it would not predict 3a > 3c, cf. Tbl. 2). Such a ranking also preserves the results in the previous section, which fully relied on MaxT. Interestingly, note how MaxT values for 3b (*Long/Non Island*) and 3c (*Short/ Island*) tie here, as the additional structural complexity of 3c does not interact with the main movement dependency (*who* raising from Spec,TP to Spec,CP). Moreover, the *Short/Non Island* (3a) and *Short/Island* (3c) conditions have very similar structures (with an extraction out of the main subject). Nonetheless, the memory metrics are able to capture subtle differences in the way the parser goes through the two sentences (arguably capturing the "island construction" cost of (Kluender and Kutas, 1993)).

4.3 Adjunct and Complex NP Islands

So far, we have been successful in replicating SWP's acceptability judgments via the MG parser. However, we might wonder whether this success is due to something peculiar in the way the Subject island test cases interact with the MG parsing strategy. Thus, we tested the MG parser on Adjunct and Complex NP islands, again using as a baseline the results in SWP's Experiment 1. The test sentences for the adjunct case were as follows:

(4) a. Who *t* thinks that John left his briefcase at the office? **Short/Non Island**
b. What do you think that John left *t* at the office? **Long/Non Island**
c. Who *t* laughs if John leaves his briefcase at the office? **Short/Island**
d. What do you laugh if John leaves *t* at the office? **Long/Island**

As for Subject islands in case 2, ⟨MaxT,SumS⟩ correctly predicts the pattern of acceptability reported by SWP, matching the empirical results across all conditions (cf. Tbl. 1). Similar results are obtained for the Complex NP island, with test sentences as follows:

(5) a. Who *t* claimed that John bought a car? **Short/Non Island**
b. What did you claim that John bought *t*? **Long/Non Island**

Clause Type	Ex. #	MaxT	SumS
Short/Non Island	4a	13/*PP*	10
Long/Non Island	4b	17/*PP*	18
Short/Island	4c	13/*PP*	11
Long/Island	4d	21/*PP*	28
Short/Non Island	5a	5/*C*	9
Long/Non Island	5b	13/*did*	19
Short/Island	5c	5/*C*	9
Long/Island	5d	15/*did*	21

Table 3: Adjunct Island and Complex NP Island: MaxT (*value/node*) and SumS values by test sentence.

c. Who *t* made the claim that John bought a car? **Short/Island**
d. What did you make the claim that John bought *t*? **Long/Island**

Once more, the parser matches the acceptability preferences reported in SPW correctly in all conditions. Particularly interesting is the absence of a contrast between 4a and 4c. This is again due to the absence of a real interaction between the additional structural complexity of the island and the main movement dependency. The fact that this results in a tie stresses how movement dependencies and structural complexity conspire with the top-down strategy of the MG parser in non-trivial ways to drive memory cost.

5 Discussion

This paper argues for an MG parser as a good, non probabilistic formal model of how gradient acceptability can be derived from categorical grammars. In doing so, we provide one of the first quantitative models of how processing factors and fine-grained, minimalist-like grammatical information can conspire to modulate acceptability. As a proof-of-concept, we replicated the gradient acceptability scores for the island effects in (Sprouse et al., 2012a). These results are certainly preliminary, but the success of the parser on this baseline is encouraging.

As mentioned in the Introduction, many hypotheses have been formulated in the past about the way memory and grammatical factors conspire to produce processing differences across sentences. Thus, it is reasonable to wonder what are the benefits of the particular linking hypothesis implemented here. As we pointed out before, one of the main advantages of our model is the tight connection between the parser behavior and the rich grammatical information encoded in the MG derivation trees. This allows for rigorous evaluations of the cognitive claims made by modern syntactic theories.

In line with recent work using the MG parser as a model of processing difficulty, Section 4 focused on the predictions made by MAXT and SUMS, Clearly, one could easily conceive of metrics that take different syntactic information into account (for example, by counting the amount of bounding nodes or phases). However, tenure and size arguably rely on the simplest possible connection between memory, structure, and parsing behavior — as they exclusively refer to the geometry of a derivation tree, without additional assumptions about the nature of its nodes.

Of course, a question remains about the cognitive plausibility of such metrics. While this model is certainly not the first to formalize memory cost as associated to the length of movement dependencies, the previous discussion highlighted how size-centered metrics do not simply depend on the length of a movement steps. Instead, they pick up on the non-trivial changes in the behavior of the parser, based on how long-distance dependencies interact with local structural configurations. Thus, they cannot trivially be identified with other length-based measures (cf. Gibson, 1998; Rambow and Joshi, 2015, a.o.). As previous work points out, in the future it will be important to explore the relation between these complexity metrics, and psychological insights about the nature of human memory mechanisms (De Santo, 2019).

Similarly, as one reviewer suggests, it would be interesting to see whether SPW's results can be derived from different cognitive hypotheses; for instance by implementing in the MG model the variety of constraints explored by Boston (2012) for a dependency parser. Moreover, in this study we employ a deterministic parser to exclusively focus on the relation between structural complexity and memory usage. However, it is known that structural and lexical frequency influence islands' acceptability (Chaves and Dery, 2019, a.o.). Thus, informative insights would come from implementing information-theoretical complexity metrics over the MG parser (Hale, 2016), and explore the predictions of expectation-based approaches.

Obviously, the target judgments modeled here are part of a restricted set. Future studies in this sense will benefit from wider comparisons among minimally different variants of acceptable and unacceptable sentences (cf. Sprouse et al., 2013, 2016). As mentioned, the nature of the model makes comparisons beyond pairs of minimal sentences hard to interpret. However, in future it might be possible to define normalization measures for memory metrics computed over sentences with widely different underlying structures.

Finally, in Section 3 we avoided discussing the nature of island effects, as we do not mean for the MG model to address the debate of whether island violations are reducible to processing factors, or are instead tied to core grammatical constraints. Importantly, while this approach might superficially be construed as a reductionist theory, it is not: for instance, the MG parser by itself is not able to explain the difference between sentences that are simply hard to process, and sentences considered unacceptable/ungrammatical. Thus, the model is theoretically neutral with respect to grammatical or reductionist frameworks.

However, consider the first case of Subject islands we analyzed in Sec. 4. The parser produced the right predictions for all test sentences except when, in the presence of an island construction, the longest movement dependency and the island violation did not coincide (2c and 2d). This mismatch is not only explained, but it is actually expected, if we embrace a grammatical theory of island constraints. Under such theory, 2d is preferable from a processing perspective (as it involves shorter dependencies), but its acceptability is lowered by the fact that it violates a grammatical constraint, while 2c does not.

While we have to be careful in formulating hypotheses based on a single data point, this contrast suggests that the MG model could help us investigate those aspects of acceptability that are fundamentally tied to grammatical constraints.

Acknowledgements

Sincere thanks to Thomas Graf, Jon Sprouse, and three anonymous reviewers for insightful feedback on this work.

References

David Adger. 2003. *Core syntax: A minimalist approach*, volume 33. Oxford: Oxford University Press.

Marisa Ferrara Boston. 2010. The role of memory in superiority violation gradience. In *Proceedings* *of the 2010 Workshop on Cognitive Modeling and Computational Linguistics*, pages 36–44. Association for Computational Linguistics.

Marisa Ferrara Boston. 2012. *A computational model of cognitive constraints in syntactic locality*. Ph.D. thesis, Cornell University.

Rui P Chaves and Jeruen E Dery. 2019. Frequency effects in subject islands. *Journal of Linguistics*, 55(3):475–521.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, volume 11. MIT Press.

Matthew W Crocker and Frank Keller. 2005. Probabilistic grammars as models of gradience in language processing. In *Gradience in grammar: Generative perspectives*. Citeseer.

Aniello De Santo. 2019. Testing a Minimalist grammar parser on Italian relative clause asymmetries. In *Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL) 2019*, June 6 2019, Minneapolis, Minnesota.

Aniello De Santo and Nazila Shafiei. 2019. On the structure of relative clauses in Persian: Evidence from computational modeling and processing effects. Talk at the 2nd North American Conference in Iranian Linguistics (NACIL2), April 19-21 2019, University of Arizona.

Sabrina Gerth. 2015. *Memory Limitations in Sentence Comprehension: A Structural-based Complexity Metric of Processing Difficulty*, volume 6. Universitätsverlag Potsdam.

E Gibson. 2000. The dependency locality theory: a distance-based theory of linguistic complexity. In *2000, Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*, pages 95–126. MIT press.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Thomas Graf, Brigitta Fodor, James Monette, Gianpaul Rachiele, Aunika Warren, and Chong Zhang. 2015. A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 1–14, Chicago, USA. Association for Computational Linguistics.

Thomas Graf and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the 2014 ACL Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36.

Thomas Graf, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for Minimalist parsing. *Journal of Language Modelling*, 5:57–106.

John Hale. 2016. Information-theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412.

Henk Harkema. 2001. A characterization of minimalist languages. In *International Conference on Logical Aspects of Computational Linguistics*, pages 193–211. Springer.

Philip Hofmeister, Laura Staum Casasanto, and Ivan A Sag. 2012a. How do individual cognitive differences relate to acceptability judgments? A reply to Sprouse, Wagers, and Phillips. *Language*, pages 390–400.

Philip Hofmeister, Laura Staum Casasanto, and Ivan A Sag. 2012b. Misapplying working-memory tests: A reductio ad absurdum. *Language*, 88(2):408–409.

Frank Keller. 2000. *Gradience in grammar: Experimental and computational aspects of degrees of grammaticality*. Ph.D. thesis.

Robert Kluender and Marta Kutas. 1993. Subjacency as a processing phenomenon. *Language and cognitive processes*, 8(4):573–633.

Gregory M Kobele, Sabrina Gerth, and John Hale. 2013. Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.

Annika Kohrt, Trey Sorensen, and Dustin A Chacón. 2018. The real-time status of semantic exceptions to the adjunct island constraint. In *Proceedings of WECOL 2018: Western Conference on Linguistics*.

Dave Kush, Terje Lohndal, and Jon Sprouse. 2018. Investigating variation in island effects. *Natural language & linguistic theory*, 36(3):743–779.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2014. Measuring gradience in speakers' grammaticality judgements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2015. Unsupervised prediction of acceptability judgements. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 1618–1628.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

So Young Lee. 2018. A minimalist parsing account of attachment ambiguity in English and Korean. *Journal of Cognitive Science*, 19(3):291–329.

Richard L Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419.

Lei Liu. 2018. Minimalist Parsing of Heavy NP Shift. In *Proceedings of PACLIC 32 The 32nd Pacific Asia Conference on Language, Information and Computation*, The Hong Kong Polytechnic University, Hong Kong SAR.

Brian McElree, Stephani Foraker, and Lisbeth Dyer. 2003. Memory structures that subserve sentence comprehension. *Journal of memory and language*, 48(1):67–91.

Jens Michaelis. 1998. Derivational minimalism is mildly context–sensitive. In *International Conference on Logical Aspects of Computational Linguistics*, pages 179–198. Springer.

Alan Prince and Paul Smolensky. 2008. *Optimality Theory: Constraint interaction in generative grammar*. John Wiley & Sons.

Owen Rambow and Aravind K Joshi. 2015. A processing model for free word-order languages. *Perspectives on sentence processing*.

Luigi Rizzi. 1990. *Relativized minimality.* The MIT Press.

J. R. Ross. 1968. Constraints on variables in syntax. *Ph.D. dissertation, MIT*.

Antonella Sorace and Frank Keller. 2005. Gradience in linguistic data. *Lingua*, 115(11):1497–1524.

Jon Sprouse. 2007. Continuous acceptability, categorical grammaticality, and experimental syntax. *Biolinguistics*, 1:123–134.

Jon Sprouse, Ivano Caponigro, Ciro Greco, and Carlo Cecchetto. 2016. Experimental syntax and the variation of island effects in english and italian. *Natural Language & Linguistic Theory*, 34(1):307–344.

Jon Sprouse, Carson T Schütze, and Diogo Almeida. 2013. A comparison of informal and formal acceptability judgments using a random sample from linguistic inquiry 2001–2010. *Lingua*, 134:219–248.

Jon Sprouse, Matt Wagers, and Colin Phillips. 2012a. A test of the relation between working-memory capacity and syntactic island effects. *Language*, 88(1):82–123.

Jon Sprouse, Matt Wagers, and Colin Phillips. 2012b. Working-memory capacity and island effects: A reminder of the issues and the facts. *Language*, 88(2):401–407.

Jon Sprouse, Beracah Yankama, Sagar Indurkhya, Sandiway Fong, and Robert C Berwick. 2018. Colorless green ideas do sleep furiously: Gradient acceptability and the nature of the grammar. *The Linguistic Review*, 35(3):575–599.

Edward P Stabler. 1997. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer.

Edward P Stabler. 2011. Computational perspectives on minimalism. In *The Oxford Handbook of Linguistic Minimalism*.

Edward P Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5(3):611–633.

Robert Truswell. 2011. *Events, phrases, and questions*. 33. Oxford University Press.

Heinz Wanner and Michael P. Maratsos. 1978. An ATN approach to comprehension. In *Linguistic theory and psychological reality*. MIT Press.

Victor H Yngve. 1960. A model and an hypothesis for language structure. *Proceedings of the American philosophical society*, 104(5):444–466.

Chong Zhang. 2017. *Stacked Relatives: Their Structure, Processing and Computation*. Ph.D. thesis, State University of New York at Stony Brook.

Evolving constraints and rules in Harmonic Grammar*

Elliott Moreton
University of North Carolina, Chapel Hill
moreton@unc.edu

Abstract

An evolutionary model of pattern learning in the MaxEnt OT/HG framework is described in which constraint induction and constraint weighting are consequences of reproduction with variation and differential fitness. The model is shown to fit human data from published experiments on both unsupervised phonotactic (Moreton et al., 2017) and supervised visual (Nosofsky et al., 1994) pattern learning, and to account for the observed reversal in difficulty order of exclusive-or vs. gang-effect patterns between the two experiments. Different parameter settings are shown to yield gradual, parallel, connectionist- and abrupt, serial, symbolic-like performance.

1 Introduction

Some constraints in natural-language grammars must be induced from phonological data, such as constraints which refer to specific lexemes, (e.g., McCarthy and Prince 1993; Fukazawa 1999; Pater 2000; Ota 2004; Pater 2007; Coetzee and Pater 2008; Pater 2009; Becker 2009), to specific lexical strata, inflectional paradigms, or other language-particular lexical classes, (e.g., Benua 1997; Alderete 1999; Ito and Mester 2001; Flack 2007a; Inkelas 2008), or to phonetically arbitrary sound classes that do not recur across languages (e.g., Bach and Harms 1972; Anderson 1981; Buckley 2000), as well as those which enforce idiosyncratic requirements (e.g., Prince and Smolensky 1993, 101). [1]

How and when are phonological markedness constraints induced? Proposals in the Optimality Theory/Harmonic Grammar literature fall into two main categories: *exhaustive search*, in which the learner considers all of a set of possible constraints, keeping those that best satisfy criteria (Hayes and Wilson, 2008; Wilson and Gallagher, 2018), and *error-patching*, in which the learner identifies a particular error type and makes a constraint against it (Adriaans and Kager, 2010; Pizzo, 2013; Pater, 2014).[2]

Here we discuss an alternative, *evolution*. Evolution-based algorithms are attractive because they are both an established technology for efficiently searching large, inconveniently-shaped hypothesis spaces (Bäck, 1996; Eiben and Smith, 2003; De Jong, 2006), and the basis of a leading account of human creativity in art, engineering, science, and other domains (Campbell, 1960; Simonton, 1999; Dietrich and Haider, 2015). In the specific model considered here, Winnow-MaxEnt-Subtree Breeder, constraints interact via Max Ent Harmonic Grammar (Goldwater and Johnson, 2003), but weights are population sizes, weight update is population growth or shrinkage in response to fitness-based selection, and constraints are innovated via mutation and recombination.

The paper is structured as follows. §2 describes the model (the "Winnow-MaxEnt-Subtree Breeder"). §3 illustrates some of its properties using a simplified "toy" example (Simulation 1). §4 quantifies a necessary condition for learnability in terms of the learning rate, the mutation rate, and the number of critical constraints. §§5 and 6 illustrate how the model accounts for human data from two published experiments which tested formally analogous patterns but found very different

*The author is indebted for comments and suggestions to Katya Pertsova, Jennifer Smith, participants in the UNC-Chapel Hill P-Side caucus, and three anonymous SCiL reviewers. The research was supported in part by NSF BCS 1651105, "Inside phonological learning", to E. Moreton and K. Pertsova.

[1] Constraint induction from phonetics is a separate issue, and is not addressed here; see, e.g., Hayes 1999; Smith 2002; Flack 2007b.

[2] A learner using positive rather than negative constraints can identify correct forms and make constraints that reward them (Boersma and Pater, 2007).

results, the unsupervised phonotactic learning of Moreton et al. (2017) and the supervised visual pattern learning of Nosofsky et al. (1994). Appropriate parameter settings cause the model to act in the first case more like a connectionist net (e.g., Gluck and Bower 1988b,a) and in the second case more like a serial, rule-based hypothesis-tester (e.g., Nosofsky et al. 1994; Ashby et al. 2011; Goodwin and Johnson-Laird 2013). §7 suggests further empirical tests of the model.

2 Winnow-MaxEnt-Subtree Breeder

The anatomy of Winnow-MaxEnt-Subtree Breeder will be briefly described here. It is based on a model described in Moreton (2010b,a,c) and analyzed in Moreton (2019), which it modifies and extends.[3]. Source code and a replication kit can be found at `https://users.castle.unc.edu/~moreton/Software/SCiL2020ReplicationKit/`.

2.1 Constraints and candidates

Consubstantiality of candidates and constraints. Candidates are represented using prosodic and Feature-Geometric trees familiar from existing phonological theory (Goldsmith, 1976; McCarthy, 1981; Sagey, 1990; Clements and Hume, 1995) — in this paper, a slightly simplified version of the feature system in Gussenhoven and Jacobs (2005, Ch. 5). A box marks the $\boxed{\text{head}}$; `L` and `R` mark left and right constituent boundaries. A constraint is a representational subtree, rooted at a PrWd, which describes a locus of violation (or satisfaction depending on the polarity of the constraint). Figure 1 depicts a micro-constraint that implements ONSET, à la Smith (2006). Any notational variant of this micro-constraint would belong to the same macro-constraint.

Meta-constraints. Since constraints are consubstantial with representations, they can evaluate each other. Winnow-MaxEnt-Subtree Breeder therefore allows the user to define metaconstraints, constraints which award a fitness bonus or penalty to other constraints. These can be used to prevent ill-formed constraints (e.g., *[+high][+low]), or to gently encourage or discourage constraints of particular types (e.g., those that mention "salient" features, or express particular phonetic principles).

[3]Erratum for that SCiL paper: p. 5, below Eqn. 35, "$\geq \log x$" should be "$\approx \log x$".

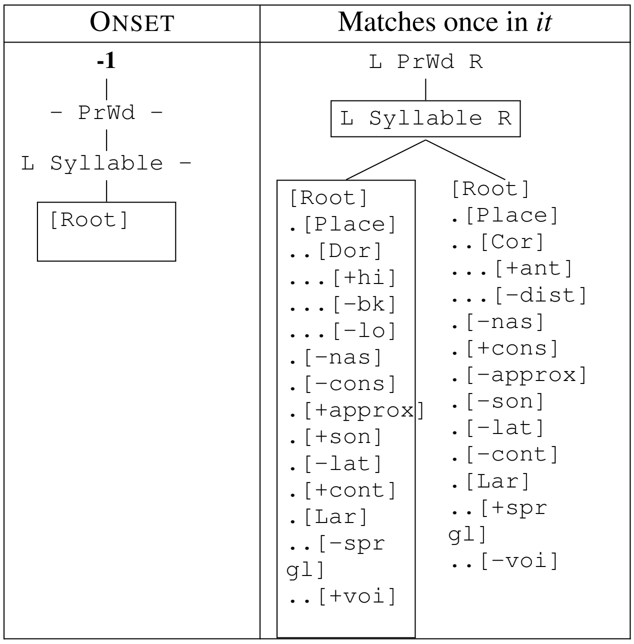


Figure 1: A constraint uses a subtree to describe a locus of violation.

2.2 Learning constraint "weights"

Weights are population sizes. In a Harmonic Grammar framework (Legendre et al., 1990), we can, without changing candidate harmonies, replace any constraint C_i of weight w_i with w_i/ζ constraints that each contribute ζ to harmony. For example, if $\zeta = 0.01$, a MAX constraint of weight 3.5 can be replaced by 350 micro-MAX's, each of which has weight 1 and whose marks are multiplied by 0.01. In Winnow-MaxEnt-Subtree Breeder, all constraints are micro-constraints of fixed weight 1. The harmony of a candidate x is

$$h(x) = \sum_c \zeta c(x) \tag{1}$$

Luce/MaxEnt choice rule. Given the experimenter's intended winner x^+ and intended loser x^-, the learner chooses x^+ with a probability that depends on the harmonies of the candidates.

$$\Pr(x^+ \mid x^+, x^-) = \frac{e^{h(x^+)}}{e^{h(x^+)} + e^{h(x^-)}} \tag{2}$$

This is the two-alternative Luce choice rule (Luce, 1959, 23) applied to the exponentiated harmonies, i.e., a conditional Maximum Entropy model (Goldwater and Johnson, 2003; Jäger, 2007; Hayes and Wilson, 2008). The generalization to k alternatives is straightforward. The total harmony available in the system is thus $N\zeta$, limiting performance.

Macro-constraints. The algorithm itself is cognizant only of micro-constraints. For analytic

d_i	Favors	Expected offspring o_i	Effect on population of $[c_i]$
-1	loser	$1/(1+\eta) < 1$	shrinks
0	neither	1	stays same
$+1$	winner	$1+\eta > 1$	grows

Table 1: Effect of error on offspring of micro-constraint and population of macro-constraint.

convenience, we, looking in from outside, can classify two micro-constraints c_i, c_j as belonging to the same *macro-constraint* if they assign the same scores to all candidates in the representational space. In the example above, the 350 micro-MAX's belong to a macro-constraint with a population size of 350 and an effective weight of 3.5. Macro-constraint membership is an equivalence relation, so we can write $[c_i]$ for the macro-constraint containing the micro-constraint c_i.

Weight update is reproduction. When an error occurs, each micro-constraint c_i produces an expected number of offspring given by

$$o_i = (1+\eta)^{d_i} \tag{3}$$

where η is a learning-rate parameter and $d_i = c_i(x^+) - c_i(x^-)$ is the difference between the winner's and loser's score on c_i. (The quantity o_i is the *fitness* of c_i.) In particular, c_i produces $\lfloor o_i \rfloor$ offspring with certainty, and one more with probability $o_i - \lfloor o_i \rfloor$. E.g., if c_i is binary (awards 0 or 1 marks), then Table 1 shows the expected number of offspring of the micro-constraint and the effect on the population size of the macro-constraint. This update rule induces a variant of the Winnow-2 algorithm (Littlestone, 1988; Moreton, 2019), first mentioned as a possible HG learning algorithm by Magri (2013).

If "soft" meta-constraints (those that assign finite marks) are specified, they add an offset to o_i equal to ζ times the total score they assign to c_i.

2.3 Evolving constraints

The initial constraint population is set by the user. Thereafter, on each error, the population is completely replaced via the following procedure.

Breeding. For each micro-constraint c_i in the pre-error population P, $o_i \cdot s$ identical clone offspring are made and deposited in the reproductive population R. Here s is the "clutch size" parameter, 1 by default, which allows the absolute number of offspring to be varied while maintaining the relative proportions belonging to differently-fit parents.

Recombination. Of the constraints in R, $\lfloor \sigma \cdot |R| + 0.5 \rfloor$ are randomly selected to be *recombinant* breeders, partitioning R into B (recombinant breeders) and $R - B$ (parthenogenetic breeders). The offspring population O is initialized to equal $R - B$. For each breeder $c_i \in B$, another breeder $c_j \in B$ of equal or greater fitness ($o_i \leq o_j$) is selected, and the two constraints are combined as described in Moreton (2019) to make a new constraint $c_{i,j}$, which is then added to O. (Recombination is not used in the simulations described in this paper.)

Mutation. Of the constraints in O, $\lfloor \mu |O| + 0.5 \rfloor$ are randomly selected to undergo mutation. Mutation is undirected, i.e., the probability of a particular mutation is independent of its effect on fitness (just as in the Minimum Description Length learner of Rasin and Katzir 2016). Mutation proceeds recursively, starting with the highest node in the constraint. Mutation operations differ between node genera (Table 2). At each node, every operation that can apply to that node first has a chance to apply. Then the algorithm visits each actual *or potential* dependent of the node, and applies recursively to it. A potential dependent of a unary feature is any currently unrealized dependent feature; e.g., an unfilled [ant] slot under [+Cor]. A potential dependent of a prosodic category is an interval between two of its actual constituents, counting the category's own boundaries as constituents. For example, the PrWd in $[\sigma\sigma]_{\text{PrWd}}$ has two actual dependents (the two σs) and three potential ones: $[\smile\sigma\smile\sigma\smile]_{\text{PrWd}}$. Mutation could add another σ node at any or all of the three potential dependents.

After mutation has applied to a constraint, the mutant and the original are compared, and if they are identical, or if the mutant receives marks from a "hard" meta-constraint (one that assigns marks of $-\infty$), mutation is re-attempted until an actual mutant is achieved. The number of mutants produced on each error is thus $Ns\mu$.

The probability of each operation can be set individually. In the present simulations, all are set to the same probability π, except those for *Gain head*, *Lose head*, and *Duplicate constituent*, which are set to 0. The larger π is, the more the mutant will differ from the parent.

A micro-constraint which is lost from the population and later re-innovated returns with its old fitness value, rather than the default fitness of 1 given to novel micro-constraints. (This design choice is

Invert polarity: Change the sign of the mark given by a constraint.

Add constituent: Applied to a potential dependent in a PrWd (syllable), adds a syllable node (segment node) there. (E.g., $[\sigma\sigma]_{\text{PrWd}}$ has three potential dependents, marked here with $\smile$s: $[\smile\sigma\smile\sigma\smile]_{\text{PrWd}}$. Each $\smile$ could mutate into another syllable.)

Delete constituent: Applied to a syllable node (segment node), deletes it.

Duplicate constituent: Applied to a syllable or segment, makes an adjacent duplicate copy of the syllable or segment, including all of its dependents.

Gain head: Applied to a PrWd (syllable), designates one of its syllables (segments) as the head, or moves the head if there already is one.

Lose head: Applied to a PrWd (syllable), makes it headless by undesignating the existing head (if any)

Flip anchor: Applied to a prosodic boundary marker, toggles it (between `-` and `L`, or between `-` and `R`).

Gain unary: Applied to a *potential* unary feature (e.g., the empty position under a [+Place] node where [+Cor] could go), adds that unary feature.

Lose unary: Applied to an *actual* unary feature, deletes it along with all of its dependents.

Gain binary: Applied to a *potential* binary feature (e.g., the empty position under a [+Cor] node where [±ant] could go), adds that feature (with $+$ and $-$ values equally likely).

Lose binary: Applied to an *actual* binary feature, deletes it.

Invert binary coefficient: Applied to an *actual* binary feature, changes $+$ to $-$ and vice versa.

Table 2: List of mutation operations.

crucial to the success of Simulation 3 in §6.)

Memorization. With probability p_{mem}, the learner creates a new micro-constraint that gives $+1$ mark to the candidate that should have won, or -1 mark to the candidate that should not have (the experimenter can set a switch, `mem_polarity`). This constraint is cloned n_{mem} times, and the clones are added to O. (In all simulations in this paper, $p_{\text{mem}} = 0$.)

Population adjustment. The resulting offspring population is adjusted in size to meet the target size of N. The default method (*random adjustment*) is to randomly delete or clone micro-constraints, with equal probability. An alternative (*fitness-based adjustment*) is to choose the fittest N offspring, with ties broken randomly. The adjusted population then completely replaces the previous generation.

The parameters are listed in Table 3. In all the simulations reported here, the parameters were fixed across trials within a simulation, although in fact they can be varied from trial to trial.

N	Number of micro-constraints in population.
ζ	Weight quantum.
η	Learning rate.
μ	Mutation rate.
s	Clutch size.
p_{mem}	Probability to memorize winner/loser as constraint.
n_{mem}	Number of copies of winner/loser memorized.
`mem_polarity`	Memorize winner or loser?
`meta`	Meta-constraint set
`mut`	Mutation probabilities (see Table 2)
`rec`	Recombination parameters (not discussed here)

Table 3: List of simulation parameters.

3 Simulation 1: 2AFC phonological learning (toy example)

Since new macro-constraints arise by mutation out of old ones, existing macro-constraints should prime discovery of new ones that are similar to them. Since high-weighted (populous) macro-constraints initiate more mutations, new macro-constraints should tend to be mutants of (hence, similar to) high-weighted old ones. And because approximate solutions can prosper when the learner has not yet discovered the precise constraints, an approximately-right constraint can focus the learner's mutational searching on its own neighborhood.

We illustrate these general principles of the model's behavior using a stripped-down "toy" example. The stimulus space is the set of all $(C)V(C)$ where C is one of /p, t, k/ and V = /u/ . Pattern A has two place restrictions on the coda; Pattern B has one on the coda and one on the onset (Table 4).

To make analysis easier, σ is set to 0 to make all reproduction asexual (this is true througout this paper). The mutation distance between the critical constraints in Condition A is then 2 (from *[–syll, +Lab]]$_\sigma$ to *[–syll, +Dor]]$_\sigma$: delete [+Lab],

	Pattern A
Unviolated constraints	*[–syll, +Dor]]$_\sigma$ (=NoDorCoda) *[–syll, +Lab]]$_\sigma$ (=NoLabCoda)
Positive	u, ut, pu, put, tu, tut, ku, kut
Negative	up, uk, pup, puk, tup, tuk, kup, kuk
	Pattern B
Unviolated constraints	*[–syll, +Dor]]$_\sigma$ (=NoDorCoda) *$_\sigma$[[–syll, +Lab] (=NoLabOns)
Positive	u, up, ut, tu, tup, tut, ku, kup, kut
Negative	uk, pu, pup, put, tuk, kuk

Table 4: Phonotactic patterns for Simulation 1.

insert [+Dor]), while that between those in Condition B is 4 (from *[–syll, +Lab]]$_\sigma$ to *$_\sigma$[[–syll, +Dor]: delete [+Lab], insert [+Dor], unset right boundary, set left boundary). The same holds for other micro-constraints that instantiate these macro-constraints, because they likewise occur in pairs (e.g., with a useless [+nas] feature added to both). Discovering either critical constraint should therefore prime discovery of the other better in the A condition than in the B condition. Concretely, we expect that in Condition A, as compared to Condition B, (1) time between discovery of the two constraints will be smaller, and (2) the weights of the two constraints will be more strongly correlated (because they co-exist for longer).

The simulation parameters were set as follows: learning rate $\eta = 0.25$, mutation rate $\mu = 0.05$, a population of $N = 200$ constraints initialized to `*(L PrWd R)`, weight quantum $\zeta = 0.05$. The individual probabilities of the mutation operations *Add constituent, Delete constituent, Flip anchor, Gain unary, Lose unary, Gain Binary, Lose binary, Invert binary coefficient* were set to $\pi = 0.005$, and all the others to 0. The time limit was 1024 trials, and 100 replications of each condition were run. Non-discovery was coded as ∞, so aggregate results are reported as medians, not means.

Prediction (1): Time between discovery smaller in A than B: The median number of trials that elapsed between discovery of the two critical constraints was 2.8 times greater in Condition B than in Condition A, as shown in Table 5. The difference was significant by a Wilcoxon-Mann-Whitney rank-sum test ($U = 2657.5, p = 0.003082$, using `wilcox.test` in R's `stats` library, R Core Team 2018).

Prediction (2): Weights of the two constraints more strongly correlated in A than B: Because discovery is more simultaneous in Condition A,

	Discovery of *[–syll, +Dor]]$_\sigma$	Discovery of *$_\sigma$[[–syll, +Lab] *or* *[–syll, +Lab]]$_\sigma$	Abs. diff.
A	237	243	114
B	313	316	322

Table 5: Median trials to and between discovery of critical constraints in Simulation 1, Conditions A vs. B.

the critical macro-constraints' weights develop more asymmetrically in Condition B. The mean correlation between the weights of *[–syll, +Dor] and the other critical macro-constraint was 0.72 in Condition A, 0.56 in Condition B (significantly different by a Wilcoxon-Mann-Whitney rank-sum test, $U = 4761, p = 0.0003484$. Non-discovery meant no correlation could be computed for 5 of the A and 24 of B simulations.).

Attention-like effects: Clues in the data can cause the learner to search some regions of constraint space more intensively. Here, the constraint *[–syll, +Dor]]$_\sigma$ (i.e., NoDorsalCoda, critical in A and B conditions) is one mutation away from *[–syll]]$_\sigma$ (i.e., NoCoda). The latter constraint is discovered early and simultaneously in both A and B (see Table 6). It is better supported by the training data in A (4 out of 8 positive vs. 0 out of 8 negative stimuli) than in B (3 out of 8 positive vs. 1 out of 8 negative). Once discovered, its population grows for longer in A than in B, peaking at 59 micro-constraints on Trial 305 vs. 23 micro-constraints on Trial 260. Between discovery and peak, the *[–syll]]$_\sigma$ population grew at a rate of $59/(305 - 24) = 0.21$ micro-constraints per trial in Condition A, but only $23/(259 - 24) = 0.10$ in Condition B, i.e., half as fast. More population in *[–syll]]$_\sigma$ means more opportunities to spawn *[–syll, +Dor]]$_\sigma$, and indeed that constraint is found sooner in Condition A (estimate is 72 trials by Wilcoxon-Mann-Whitney test, $U = 2657.5, p = 0.003082$). Across all 99 replications in Condition A in which both constraints were discovered, a mean of 47% of all instances of *[–syll, +Dor]]$_\sigma$ were immediate offspring of *[–syll]]$_\sigma$. The analogous figures for Condition B are 91 and 8.7%.

Speaking anthropomorphically, we might say that the A learner "notices" that codas matter, i.e. up-weights *[–syll]]$_\sigma$. That "directs its attention" to the coda position (by allowing the approximate solution *[–syll]]$_\sigma$ to elbow out other constraints). This "focused attention" results in a more-intensive search among neighbors of *[–

syll]]$_\sigma$, which soon finds *both* critical constraints. Thus, R&D work that helps find one constraint also helps find the other. The critical constraints then outcompete the approximate constraint and drive its weight down. In the B condition, it takes longer to discover the critical constraint because the mutant population is divided between constraints targeting the onset and coda positions, i.e., the data does not "call attention" to one position more than the other.

	Median trial number	
Event	A	B
*[–syll]]$_\sigma$ discovered	24	24
*[–syll]]$_\sigma$ population peaks	305	260
(peak pop. size)	(59)	(23)
*[–syll, +Dor]]$_\sigma$ discovered	237	312

Table 6: Discovery of *[–syll]]$_\sigma$ (NOCODA) primes discovery of *[–syll, +Dor]]$_\sigma$ (NODORSALCODA) in Simulation 1.

4 Mutation, learning, and complexity in a monostratal grammar

The pattern in Simulation 1 can be captured by a monostratal grammar: The two macro-constraints handle disjoint, exhaustive subsets of the pattern, and are not critically ranked (weighted) relative to each other. In the general monostratal case, there are n critical macro-constraints in the minimal solution, with $[c_k]$ having exclusive responsibility for Trial Type k. Suppose that the learner has already found them all, and that ζ and N are big enough that growth in the population of any critical macro-constraint comes mainly at the expense of non-critical constraints (assumed to be neutral). We will see that η and μ impose an upper bound on n.

Let r_k be the probability that when the next error occurs, it will occur on Trial Type k. Then the expected proportional change in the population size w_k of $[c_k]$ is the expected product of its rates of growth through reproduction and of shrinkage through mutation. If we assume what is typically the case, that mutation turns a critical constraint into another critical constraint negligibly often, then on the next error, $[c_k]$ reproduces with probability r_k and then shrinks by mutation with probability 1:

$$\begin{aligned} E\left[w'_k/w_k\right] &= r_k(1+\eta)(1-\mu) + (1-r_k)(1-\mu) \\ &= (1+r_k\eta)(1-\mu) \end{aligned} \tag{4}$$

where w'_k is the updated w_k.

When the learning algorithm converges, $E\left[w'_k/w_k\right] \geq 1$ for all k, i.e., all of the macro-constraint weights are either constant, or else increasing at the expense of the neutral constraints. Setting $E\left[w'_k/w_k\right] = 1$ and solving for r_k yields the critical value

$$r^* = \frac{1}{\eta}\frac{\mu}{1-\mu} \tag{5}$$

If $r_k < r^*$, then $w'_k < w_k$. Hence, a necessary condition for convergence is $\forall k : r_k \geq r^*$. But since the r_k's add to 1, there must be at least one k such that $r_k \leq 1/n$. Hence a stable final grammar exists only if

$$n \leq n_{\text{crit}} = \eta\frac{1-\mu}{\mu} \tag{6}$$

In Simulation 1, η and μ were chosen so that $n_{\text{crit}} = 0.25 \cdot (1-0.05)/0.05 = 4.74 > 2 = n$, and indeed, the average proportion correct for the last 16 trials was above 0.95 in both the A and B conditions. To illustrate the effect of varying n_{crit}, the simulation was re-run with all combinations of $\eta \in \{0.1, 0.15, 0.25, 0.3\}$ and $\mu \in \{0.025, 0.05, 0.1, 0.15\}$. Figure 2 shows the results in terms of proportion correct on the last 16 trials (of 2048). For $n_{\text{crit}} > 2$, the median — indeed, the lower quartile — is never below 0.9. For n_{crit} even slightly below 2, performance drops off rapidly.

The reproduction and mutation rates thus fix an upper bound on the number of critical macro-constraints in a learnable monostratal grammar. A pattern which minimally requires more than

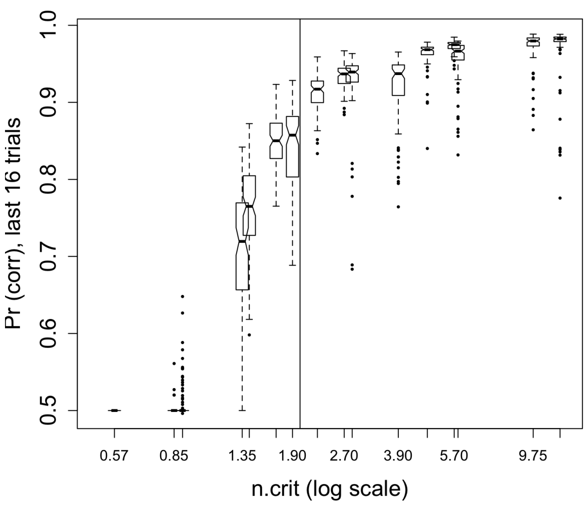


Figure 2: Proportion correct on the last 16 trials as a function of n_{crit}, Simulation 1, Condition A. Vertical line marks $n_{crit} = 2$.

n_{crit} macro-constraints cannot be learned at all. A pattern which can be expressed with n_{crit} or fewer macro-constraints cannot be learned using an equivalent monostratal grammar that has more, e.g., one relying on parochial constraints or stimulus memorization.

5 Simulation 2: Unsupervised phonological learning (Moreton et al., 2017)

When the population size N is large, and the weight quantum ζ is small, the learner approximates a constraint-based model in which the constraint set contains all possible constraints up to a certain size, whose weights vary continuously. The reason is that the mutants created on any error will sample the space of possible constraints densely. Simulation 2 illustrates this point.

In many lab experiments, phonotactic learning is *unsupervised*: Participants are trained by exposure to pattern-conforming stimuli only. Since Winnow-MaxEnt learns from winner-loser pairs, the learner must somehow generate its own loser on each trial.

A straightforward way to do that is for the learner to sample from the probability distribution specified by its current grammar. If the sample differs from the presented stimulus (virtually certain, regardless of how well the pattern has been learned), the stimulus and sample are used as x^+ and x^- in Equation 3. Since x^+ is always pattern-conforming, but x^- is sometimes not, macro-constraints enforcing the pattern prosper (i.e., gain population relative to other constraints).

The hypothesis is tested by simulating three different conditions from a published experiment (Moreton et al., 2017, Exp. 1). The stimulus space consisted of the 256 possible $C_1V_1C_2V_2$ stimuli for which the consonants were one of [t d k g] and the vowels one of [i æ u ɔ] . Human participants were familiarized by hearing and repeating aloud 32 pattern-conforming stimuli in pseudo-random order such that each stimulus occurred 4 times. They then did 32 test trials in which they heard two novel stimuli, one pattern-conforming and one not, and were asked to choose the conforming stimulus.

Three specific patterns were chosen for the simulation, each instantiating a different pattern type in the classification of Shepard et al. (1961, see Figure 3). The pattern "C_1 is voiceless" belongs to Type I, a simple, one-feature affirmation. The pattern "C_1 and C_2 disagree in voicing" is of Type II, an if-and-only-if (equivalently, an exclusive-or) combination of two features. Finally, the pattern "at least two of: (1) C_2 is velar, (2) C_1 is voiceless, (3) V_2 is back" is of Type IV, a three-feature "gang effect".

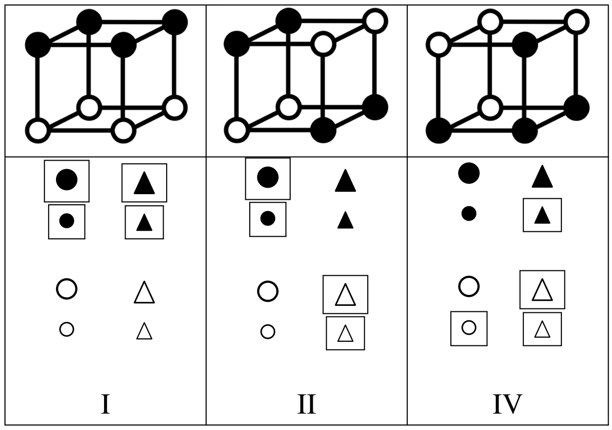


Figure 3: Pattern Types I, II, and IV of Shepard et al. (1961), illustrated using visual stimuli. Type I is defined by a single feature ("the figure is black"); Type II is an iff/xor relation between two features ("black iff round"); and Type IV is a three-feature gang effect ("at least two of white, triangular, small").

For each pattern, 32 conforming training stimuli, 32 conforming test stimuli, and 32 non-conforming test stimuli were randomly chosen. Each of the three patterns can be learned to perfection with $n = 8$ or fewer macro-constraints. The simulation parameters were set at $\eta = 0.33$, $\mu = 0.025$ (satisfying Equation 6 for $n = 8$), $\zeta = 0.05$, and $N = 2000$ constraints. The values were chosen by trial and error to approximate human performance. The test task for human participants was to decide which of each test pair was "a word in the language you were studying". In the simulation, this was implemented by attaching to each training and test stimulus a `[+real]` feature. The constraint set was initialized to equal proportions of `*[+real]` and `*[-real]`. The learner got as many training and test trials as did the humans. 100 replications of each simulation were run.

Simulation results are shown in Table 7 alongside human performance. The numbers are similar, and the proportion of pattern-conforming test-phase responses decreases in the order $I > IV > II$.

	Pattern type		
	I	II	IV
Sim.	0.83 ± 0.13	0.48 ± 0.02	0.60 ± 0.05
Human	0.73 ± 0.12	0.57 ± 0.11	0.70 ± 0.09

Table 7: Proportion pattern-conforming responses in the test phase (± 1 s.d., not s.e.m.) for Simulation 2 and human data (Moreton et al., 2017, Table 5), showing $I > IV > II$ order.

6 Simulation 3: Supervised visual learning (Nosofsky et al., 1994)

When the population size N is small and the weight quantum ζ is large, the Winnow-MaxEnt-Subtree Breeder approximates a serial hypothesis-tester that keeps trying one categorical rule after another until it finds one that works. This is illustrated in Simulation 3.

The human experiment to be replicated is that of Nosofsky et al. (1994). The stimulus space consisted of eight geometric figures varying on three binary dimensions (shape, shading, and size, as in Figure 3). A pattern was an assignment of four stimuli to Category A, and four to B. On each trial, the participant saw a figure, classified it as A or B, and received right/wrong feedback. Training continued until the participant had responded correctly on 32 consecutive trials, or reached a limit of 400 trials. The difficulty order, in terms of trials to criterion or errors to criterion, was $I < II < IV$.

Many hypothesis-testing models in the concept-learning literature account for this difficulty order by positing a bias towards syntactically-simple hypotheses (Shepard et al., 1961; Nosofsky et al., 1994; Feldman, 2006; Ashby et al., 2011; Goodwin and Johnson-Laird, 2013). The bias in Winnow-MaxEnt-Subtree Breeder has a different origin.

It can be seen from Figure 3 that a correct grammar of the Type I problem can be made with just two macro-constraints: `*[-wug][+black]` and `*[+wug][-black]`. These constraints designate the top face of the cube as a wug (i.e., pattern-conforming) and the bottom face as a non-wug. The smallest correct Type II grammar needs four constraints, one for each of the back-to-front edges of the cube (e.g., `*[-wug][+black][+circle]`). The smallest correct Type IV grammar needs six constraints, one for each of the edges radiating from the central wug or non-wug stimulus.[4] A small N should therefore favor Type I over Type II, and Type II over Type IV. For the grammar to give human-like near-categorical responses with so few constraints, the weight quantum ζ must be large, so that each constraint has the effect of a categorical rule.

The parameters for Simulation 3 were adjusted by trial and error to the values $N = 7, \zeta = 12, \eta = 1, \mu = 1, \pi = 1/2$. Clutch size was set to 12. Fitness-based selection was turned on so that the fittest N of the offspring were chosen. The high mutation rate and large clutch size should have the effect of making the offspring population be a diverse random sample of the 54 possible constraints. Any micro-constraint in the sample which has previously been seen to favor a loser will be assigned its previous (negative) fitness, and hence be eliminated from the offspring set by fitness-based selection. (Here is where the learner's memory for the fitness of extinct micro-constraints, mentioned above in §2.3, is crucial.) The result should be that, as the simulation progresses, invalid micro-constraints are gradually discovered and permanently eliminated from consideration, so that the offspring population becomes more and more a random sample of size 7 from the valid constraints.

In the Type I condition, there are 2 valid faces, 8 valid edges, and 8 valid corners, and a correct grammar can be made in many ways: from the 2 faces, from 1 face plus 4 edges, from 1 face plus 3 edges plus 2 corners, etc. In the Type II condition, there are 4 valid edges and 8 valid corners, and a correct grammar can be made from the 4 edges, or 3 edges plus 2 corners, or 2 edges plus 4 corners. In the Type IV condition, there are 6 valid edges and 8 valid corners, and a correct grammar can only be made from the 6 edges, or from 5 edges plus 2 corners, or from 2 faces plus 2 copies of each of 2 corners. Hence a random sample of size $N = 7$ is more likely to solve Type I than Type II, and Type II than Type IV.

The results of the simulation (100 replications) are shown in Table 8. The order of difficulty is the same for the learner as for the humans (who are about 40% faster in all conditions). Changing the model parameters has caused Types II and IV to change places with respect to Simulation 2. Smaller values of N amplify the advantage of

[4] Alternatively, Type IV can be expressed with two face constraints, plus two copies of each of two corner constraints to override the face constraints, which is still six constraints.

	% participants reaching criterion			Mean trials to criterion		
	I	II	IV	I	II	IV
Sim.	100	98	74	68	161	210
Human	100	100	100	44	85	127

Table 8: Attainment of criterion performance (32 consecutive correct responses in 400 trials) for Simulation 3 and human participants (Nosofsky et al., 1994, 356). Mean trials to criterion excludes cases where criterion was not reached. There were 100 replications.

Type II over Type IV. For $N \leq 5$, no Type IV simulations reach criterion.

7 Discussion

The Winnow-MaxEnt-Subtree Breeder links phonological learning theoretically with other kinds of pattern learning and with creativity in other domains, thus spawning future research questions (e.g., whether mutation is undirected, or sensitive to the demands of the problem; Simonton 1999; Dietrich and Haider 2015; whether recombination — sexual reproduction — is empirically motivated, etc.). A more immediate task is to test its empirical adequacy for phonological learning. This section suggests some places to start.

Abruptness. The learning curve in the large-N/small-ζ case is predicted to be more abrupt when the pattern depends on induced constraints rather than preexisting ones from UG or L1 (Moreton, 2019). Complex patterns require a high learning rate η and/or low mutation rate μ (see §4). Lower μ means longer intervals between constraint discoveries, while higher η means faster population growth following discoveries; hence, complex patterns are predicted to be learned as a series of sudden acquisitions of individual subpatterns. I know of no experimental evidence bearing directly on either prediction, but abruptness is a familiar aspect of first-language acquisition ("across-the-board" changes, e.g., Smith 1973; Macken and Barton 1978; Vihman and Velleman 1989; Barlow and Dinnsen 1998; Levelt and van Oostendorp 2007; Gerlach 2010; Becker and Tessier 2011; Guy 2014), and been observed in lab-learned phonology (Moreton and Pertsova, 2016). Individual learning curves for many complex non-linguistic skills show discontinuities alternating with gradual power-law improvements (Haider and Frensch, 2002; Bourne, Jr. et al., 2010; Gray and Lindstedt, 2017; Donner and Hardy, 2015).

Priming. As seen in Simulation 1, a target grammar in the large-N/small-ζ case is found sooner when the relevant macro-constraints are separated by fewer mutations, because finding one constraint generates mutants that are helpful in finding the next. The acquisition of a constraint thus primes acquisition of similar constraints. It may be relevant that, in a sample from P-Base (Mielke, 2008), Carter (2017) found that languages tend to re-use phonological features: The probability that a language which uses Feature F in N phonologically-active classes uses it in $N+1$ classes increases with N (a preferential-attachment process).

Nepotism. A weighty macro-constraint in the large-N/small-ζ case generates many mutant offspring, thereby maintaining related macro-constraints at higher weights than justified by their usefulness. Hence learners should show emergent effects of constraints that are mutationally close to high-weighted ones. In adult segment-class learning, generalization to untrained segments is stronger when they are more similar to trained segments (Cristiá et al., 2013). Prickett (2018) showed that GMECCS (a gradient-ascent Maximum Entropy learner, Pater and Moreton 2012; Moreton et al. 2017) underpredicts that difference, but that the fit can be improved by making weight updates "leak" between featurally-similar constraints. Nepotism may furnish a mechanism to cause such leakage.

Cognitive realism. Human participants report different approaches, including intuition, rote memorization, and explicit reasoning. Differences in self-reported approach correlate with differences in objective measures such as pattern-type difficulty order, learning-curve shape, and ability to verbalize the pattern (Moreton and Pertsova 2016, Moreton and Pertsova, in prep.). Simulations 2 and 3 illustrated parameter settings corresponding to intuition (large N, small ζ, random selection) and to a rudimentary sort of reasoning (small N, large ζ, fitness-based selection), and the p_{mem} parameter enables stimulus memorization. It would be desirable to know if intermediate combinations of parameter values correspond to types of human performance, how parameter values are linked to experimental conditions, and whether the number of parameters can safely be reduced.

References

Adriaans, F. and R. Kager (2010). Adding generalization to statistical learning: the induction of phonotactics from continuous speech. *Journal of Memory and Language 62*(3), 311–331.

Alderete, J. (1999). *Morphologically governed accent in Optimality Theory*. Ph. D. thesis, University of Massachusetts, Amherst.

Anderson, S. R. (1981). Why phonology isn't "natural". *Linguistic Inquiry 12*, 493–539.

Ashby, F. G., E. J. Paul, and W. T. Maddox (2011). COVIS. In E. M. Pothos and A. J. Willis (Eds.), *Formal approaches in categorization*, Chapter 4, pp. 65–87. Cambridge, England: Cambridge University Press.

Bach, E. and R. T. Harms (1972). How do languages get crazy rules? In R. P. Stockwell and R. K. S. Macaulay (Eds.), *Linguistic change and generative theory*, Chapter 1, pp. 1–21. Bloomington: Indiana University Press.

Bäck, T. (Ed.) (1996). *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford University Press.

Barlow, J. A. and D. A. Dinnsen (1998). Asymmetrical cluster development in a disordered system. *Language Acquisition 7*(1), 1–49.

Becker, M. (2009). *Phonological trends in the lexicon: the role of constraints*. Ph. D. thesis, University of Massachusetts, Amherst.

Becker, M. and A. Tessier (2011). Trajectories of faithfulness in child-specific phonology. *Phonology 28*, 163–196.

Benua, L. (1997). *Transderivational identity: phonological relations between words*. Ph. D. thesis, University of Massachusetts, Amherst, Mass.

Boersma, P. and J. Pater (2007, October). Constructing constraints from language data: the case of Canadian English diphthongs. Handout, NELS 38, University of Ottawa.

Bourne, Jr., L. E., W. D. Raymond, and A. F. Healy (2010). Strategy selection and use during classification skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition 36*(2), 500–514.

Buckley, E. (2000). On the naturalness of unnatural rules. In *Proceedings from the Second Workshop on American Indigenous Languages*, Volume 9 of *UCSB Working Papers in Linguistics*.

Campbell, D. T. (1960). Blind variation and selective retention in creative thought as in other knowledge processes. *Psychological Review 67*(6), 380–400.

Carter, W. T. (2017). Phonological activeness effects in language acquisition and language structuring. Senior Honors thesis, Department of Linguistics, University of North Carolina, Chapel Hill.

Clements, G. N. and E. V. Hume (1995). The internal organization of speech sounds. In J. A. Goldsmith (Ed.), *The handbook of phonological theory*, Chapter 7, pp. 245–306. Boston: Blackwell.

Coetzee, A. W. and J. Pater (2008). Weighted constraints and gradient restrictions on place cooccurrence in muna and arabic. *Natural Language & Linguistic Theory 26*(2), 289–337.

Cristiá, A., J. Mielke, R. Daland, and S. Peperkamp (2013). Similarity in the generalization of implicitly learned sound patterns. *Laboratory Phonology 4*, 259–285.

De Jong, K. A. (2006). *Evolutionary computation: a unified approach*. Cambridge, Massachusetts: MIT Press.

Dietrich, A. and H. Haider (2015). Human creativity, evolutionary algorithms, and predictive representations: the mechanics of thought trials. *Psychonomic Bulletin and Review 22*, 897–915.

Donner, Y. and J. L. Hardy (2015). Piecewise power laws in individual learning curves. *Psychonomic Bulletin and Review 22*(5), 1308–1319.

Eiben, A. E. and J. E. Smith (2003). *Introduction to evolutionary computing*. Berlin: Springer.

Feldman, J. (2006). An algebra of human concept learning. *Journal of mathematical psychology 50*, 339–368.

Flack, K. (2007a). Templatic morphology and indexed markedness constraints. *Linguistic Inquiry 38*(4), 749–758.

Flack, K. G. (2007b). *The sources of phonological markedness*. Ph. D. thesis, University of Massachusetts, Amherst.

Fukazawa, H. (1999). *Theoretical implications of OCP effects on features in Optimality Theory*. Ph. D. thesis, University of Maryland, College Park.

Gerlach, S. R. (2010). *The acquisition of consonant feature sequences: harmony, metathesis, and deletion patterns in phonological development*. Ph. D. thesis, University of Minnesota.

Gluck, M. A. and G. H. Bower (1988a). Evaluating an adaptive network model of human learning. *Journal of Memory and Language 27*, 166–195.

Gluck, M. A. and G. H. Bower (1988b). From conditioning to category learning: an adaptive network model. *Journal of Experimental Psychology: General 117*(3), 227–247.

Goldsmith, J. A. (1976). *Autosegmental phonology*. Ph. D. thesis, Massachusetts Institute of Technology.

Goldwater, S. J. and M. Johnson (2003). Learning OT constraint rankings using a maximum entropy model. In J. Spenader, A. Erkisson, and O. Dahl (Eds.), *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, pp. 111–120.

Goodwin, G. P. and P. N. Johnson-Laird (2013). The acquisition of Boolean concepts. *Trends in Cognitive Sciences 17*(3), 128–133.

Gray, W. D. and J. K. Lindstedt (2017). Plateaus, dips, and leaps: where to look for inventions and discoveries during skilled performance. *Cognitive Science 41*, 1838–1870.

Gussenhoven, C. and H. Jacobs (2005). *Understanding phonology* (2nd ed.). Understanding Language Series. London: Hodder Arnold.

Guy, G. R. (2014). Linking usage and grammar: generative phonology, exemplar theory, and variable rules. *Lingua 142*, 57–65.

Haider, H. and P. A. Frensch (2002). Why aggregated learning follows the power law of practice when individual learning does not: comment on Rickard (1997, 1999), Delaney et al. (1998), and Palmeri (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition 28*(2), 392–406.

Hayes, B. (1999). Phonetically driven phonology: the role of optimality in inductive grounding. In M. Darnell, E. Moravcsik, M. Noonan, F. Newmeyer, and K. Wheatly (Eds.), *Functionalism and Formalism in Linguistics*, Volume 1: General Papers, pp. 243–285. Amsterdam: John Benjamins.

Hayes, B. and C. Wilson (2008). A Maximum Entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry 39*(3), 379–440.

Inkelas, S. (2008). The morphology-phonology connection. In *Proceedings of the Berkeley Linguistics Society*, Volume 34, Berkeley, California, pp. 145–162. Berkeley Linguistics Society and Linguistic Society of America.

Ito, J. and A. Mester (2001). Covert generalizations in Optimality Theory: the role of stratal faithfulness constraints. *Studies in Phonetics, Phonology, and Morphology 7*, 3–33.

Jäger, G. (2007). Maximum Entropy models and Stochastic Optimality Theory. In J. Grimshaw, J. Maling, C. Manning, J. Simpson, and A. Zaenen (Eds.), *Architectures, rules, and preferences: a festschrift for Joan Bresnan*, pp. 467–479. Stanford, California: CSLI Publications.

Legendre, G., Y. Miyata, and P. Smolensky (1990). Can connectionism contribute to syntax? Harmonic Grammar, with an application. In M. Ziolkowski, M. Noske, and K. Deaton (Eds.), *Proceedings of the 26th Regional Meeting of the Chicago Linguistic Society*, Chicago, pp. 237–252. Chicago Linguistic Society.

Levelt, C. and M. van Oostendorp (2007). Feature co-occurrence constraints in L1 acquisition. *Linguistics in the Netherlands 24*(1), 162–172.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning 2*, 285–318.

Luce, R. D. (2005 [1959]). *Individual choice behavior: a theoretical analysis*. New York: Dover.

Macken, M. A. and D. Barton (1978, March). The acquisition of the voicing contrast in English: a study of voice-onset time in word-initial stop consonants. Report from the Stanford Child Phonology Project.

Magri, G. (2013). HG has no computational advantages over OT: toward a new toolkit for computational OT. *Linguistic Inquiry 44*(4), 569–609.

McCarthy, J. J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry 12*, 373–418.

McCarthy, J. J. and A. M. Prince (1993). Generalized alignment. In G. Booij and J. van Marle (Eds.), *Yearbook of morphology 1993*, pp. 79–153. Kluwer.

Mielke, J. (2008). *The emergence of distinctive features*. Oxford, England: Oxford University Press.

Moreton, E. (2010a, April). Connecting paradigmatic and syntagmatic simplicity bias in phonotactic learning. Department colloquium, Department of Linguistics, MIT.

Moreton, E. (2010b, February). Constraint induction and simplicity bias. Talk given at the Workshop on Computational Modelling of Sound Pattern Acquisition, University of Alberta.

Moreton, E. (2010c, May). Constraint induction and simplicity bias in phonotactic learning. Handout from a talk at the Workshop on Grammar Induction, Cornell University.

Moreton, E. (2019). Constraint breeding during on-line incremental learning. In *Proceedings of the Society for Computation in Linguistics*, Volume 2, pp. Article 9.

Moreton, E., J. Pater, and K. Pertsova (2017). Phonological concept learning. *Cognitive Science 41*(1), 4–69.

Moreton, E. and K. Pertsova (2016). Implicit and explicit processes in phonotactic learning. In TBA (Ed.), *Proceedings of the 40th Boston University Conference on Language Development*, Somerville, Mass., pp. TBA. Cascadilla.

Nosofsky, R. M., M. A. Gluck, T. J. Palmeri, S. C. McKinley, and P. Gauthier (1994). Comparing models of rule-based classification learning: a replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory and Cognition 22*(3), 352–369.

Nosofsky, R. M., T. J. Palmeri, and S. C. McKinley (1994). Rule-plus-exception model of classification learning. *Psychological Review 101*(1), 53–79.

Ota, M. (2004). The learnability of the stratified phonological lexicon. *Journal of Japanese Linguistics 20*, 19–40.

Pater, J. (2000). Non-uniformity in English secondary stress: the role of ranked and lexically specific constraints. *Phonology 17*, 237–274.

Pater, J. (2007). The locus of exceptionality: morpheme-specific phonology as constraint indexation. In L. Bateman, M. O'Keefe, E. Reilly, and A. Werle (Eds.), *Papers in Optimality Theory III*, pp. 259–296. Amherst: Graduate Linguistics Students Association, University of Massachusetts.

Pater, J. (2009). Morpheme-specific phonology: constraint indexation and inconsistency resolution. In S. Parker (Ed.), *Phonological argumentation: essays on evidence and motivation*, pp. 1–33. London: Equinox.

Pater, J. (2014). Canadian Raising with language-specific weighted constraints. *Language 90*(1), 230–240.

Pater, J. and E. Moreton (2012). Structurally biased phonology: complexity in learning and typology. *Journal of the English and Foreign Languages University, Hyderabad 3*(2), 1–44.

Pizzo, P. (2013, January 19). Learning phonological alternations with online constraint induction. Slides from a presentation at the 10th Old World Conference on Phonology (OCP 10).

Prickett, B. (2018). Similarity-based phonological generalization. In G. Jarosz and J. Pater (Eds.), *Proceedings of the Society for Computation in Linguistics*, Volume 1, pp. Article 24.

Prince, A. and P. Smolensky (1993). *Optimality Theory: constraint interaction in generative grammar*. Department of Linguistics, Rutgers University.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rasin, E. and R. Katzir (2016). On evaluation metrics in optimality theory. *Linguistic Inquiry 47*(2), 235–282.

Sagey, E. (1990). *The representation of features in non-linear phonology: the Articulator Node Hierarchy*. New York: Garland.

Shepard, R. N., C. L. Hovland, and H. M. Jenkins (1961). Learning and memorization of classifications. *Psychological Monographs 75*(13, Whole No. 517).

Simonton, D. K. (1999). Creativity as blind variation and selective retention: is the creative process Darwinian? *Psychological Inquiry 10*(4), 309–328.

Smith, J. L. (2002). *Phonological augmentation in prominent positions*. Ph. D. thesis, University of Massachusetts, Amherst.

Smith, J. L. (2006). Representational complexity in syllable structure and its consequences for Gen and Con. MS, Department of Linguistics, University of North Carolina, Chapel Hill. ROA-800.

Smith, N. V. (1973). *The acquisition of phonology: a case study*. Cambridge, England: Cambridge University Press.

Vihman, M. M. and S. Velleman (1989). Phonological reorganization: a case study. *Language and Speech 32*, 149–170.

Wilson, C. and G. Gallagher (2018). Accidental gaps and surface-based phonotactic learning: A case study of South Bolivian Quechua. *Linguistic Inquiry 49*(3), 610–623.

Interpreting Verbal Irony: Linguistic Strategies and the Connection to the Type of Semantic Incongruity

Debanjan Ghosh[1], Elena Musi[2], Kartikeya Upasani[3], Smaranda Muresan[4]
[1]McGovern Institute for Brain Research, MIT, Cambridge, MA
[2]University of Liverpool, Liverpool, UK
[3]Facebook Conversational AI, CA
[4] Data Science Institute, Columbia University, New York, NY
dg513@mit.edu, elena.musi@liverpool.ac.uk,
kart@fb.com, smara@columbia.edu

Abstract

Human communication often involves the use of verbal irony or sarcasm, where the speakers usually mean the opposite of what they say. To better understand how verbal irony is expressed by the speaker and interpreted by the hearer we conduct a crowdsourcing task: given an utterance expressing verbal irony, users are asked to verbalize their interpretation of the speaker's ironic message. We propose a typology of linguistic strategies for verbal irony interpretation and link it to various theoretical linguistic frameworks. We design computational models to capture these strategies and present empirical studies aimed to answer three questions: (1) what is the distribution of linguistic strategies used by hearers to interpret ironic messages?; (2) do hearers adopt similar strategies for interpreting the speaker's ironic intent?; and (3) does the type of semantic incongruity in the ironic message (explicit vs. implicit) influence the choice of interpretation strategies by the hearers?

1 Introduction

It is well understood that recognizing whether a speaker is ironic or sarcastic is essential to understanding their actual sentiments and beliefs. For instance, the utterance "pictures of holding animal carcasses are so *flattering*" is an expression of verbal irony, where the speaker has a negative sentiment towards "pictures of holding animal carcasses", but uses the positive sentiment word "flattering". This inherent characteristic of verbal irony is called semantic incongruity — incongruity between the literal evaluation and the context (e.g., between the positive sentiment words and the negative situation in this example). Most NLP research on verbal irony or sarcasm has focused on the task of *sarcasm detection* treating it as a binary classification task using either the utterance in isolation or adding contextual information such as conversation context, author context, visual context, or cognitive features (Davidov et al., 2010; Maynard and Greenwood, 2014; Wallace et al., 2014; Joshi et al., 2015; Bamman and Smith, 2015; Muresan et al., 2016; Amir et al., 2016; Mishra et al., 2016; Ghosh and Veale, 2017; Felbo et al., 2017; Ghosh et al., 2017; Hazarika et al., 2018; Tay et al., 2018; Ghosh et al., 2018; Oprea and Magdy, 2019). Such approaches have focused their analysis on the speakers' beliefs and intentions for using irony (Attardo, 2000). However, sarcasm and verbal irony are types of interactional phenomena with specific perlocutionary effects on the hearer (Haverkate, 1990). Thus, we argue that, besides recognizing the speaker's sarcastic/ironic intent, it is equally important to understand *how the hearer interprets* the speaker's sarcastic/ironic message. For the above utterance, the strength of negative sentiment perceived by the hearer depends on whether they interpret the speaker's actual meaning as "picture ... are **not flattering**" vs. "pictures ... are **so gross**" (Table 1). The intensity of negative sentiment is higher in the latter interpretation than in the former. Kreuz (2000) noted that most studies in linguistics and psychology have conducted experiments analyzing reaction times (Gibbs, 1986; Katz et al., 2004) or situational context (Ivanko and Pexman, 2003), featuring a setup with *in vitro* data aimed at testing the validity of specific theories of irony. Instead, our study adopts a *naturalistic* approach to understand hearers' reception of irony looking at what linguistic strategies are recurrently used by hearers to interpret the non-literal meaning underlying ironic utterances.

We leverage the crowdsourcing task introduced by Ghosh et al. (2015) for their work on detecting whether a word has a literal or sarcastic in-

[1]Part of the research was carried out while Debanjan was a Ph.D. candidate at Rutgers University.

terpretation, later adopted by Peled and Reichart (2017). The task is framed as follows: given a speaker's ironic message, five annotators (e.g., Turkers on Amazon Mechanical Turk (MTurk)) are asked to verbalize their interpretation of the speaker's ironic message (i.e., their understanding of the speaker's intended meaning) (see Table 1; S_{im} denotes the speaker's ironic message, while H_{int} denotes the hearer's interpretation of that ironic message). The crowdsourcing experiments are reported in Section 2.

The paper makes three contributions. First, we propose a data-driven *typology of linguistic strategies* that hearers use to interpret ironic messages and discuss its relevance in verifying theoretical frameworks of irony (Section 4). Second, we propose computational models to capture these strategies (Section 5). Third, we present two studies that aim to answer two questions: (1) does the type of semantic incongruity in the ironic message (explicit vs. implicit; see Section 3) influence the choice of interpretation strategies by the hearers? (Section 6.2); (2) do interpretation strategies of verbal irony vary by hearers? We make all datasets and code available.[1]

2 Datasets of Speakers' Ironic Messages and Hearers' Interpretations

To generate a parallel dataset of speakers' ironic messages and hearers' interpretations we conduct a crowdsourcing experiment. Given a speaker's ironic message (S_{im}), five Turkers (hearers) on MTurk are asked to verbalize their interpretation of the speaker's ironic message (i.e., their understanding of the speaker's intended meaning) (H_{int}). The design of the MTurk task was first introduced by Ghosh et al. (2015), who use the resulting dataset to identify words that can have both a literal and a sarcastic sense. Peled and Reichart (2017) employed similar design to generate a parallel dataset to use for generating interpretations of sarcastic messages using machine translation approaches. They use skilled annotators in comedy writing and literature paraphrasing and give them the option not to rephrase (we refer to Peled and Reichart (2017)'s dataset as $SIGN$). We perform this new crowdsourcing task and do not rely entirely on the above two datasets for two reasons: (1) we focus on verbal irony, and (2) we always require an interpretation from the Turkers. Unlike the above two studies, the main goal of our research is to analyze the linguistics strategies employed by hearers in interpreting verbal irony.

We collected messages that express verbal irony from Twitter using the hashtags #irony, #sarcastic, and #sarcasm. We chose Twitter as a source since the presence of the hashtags allows us to select sentences where the speaker's intention was to be ironic. Furthermore, even though Twitter users cannot be considered representative of the entire population, they are unlikely to be skewed with respect to topics or gender. We manually checked and kept 1,000 tweets that express verbal irony. We do not draw any theoretical distinction between sarcasm and irony since we cannot assume that Twitter users also differentiate between #irony and #sarcasm, blurred even in scholarly literature. The Turkers were provided with detailed instructions and examples of the task including the standard definition of verbal irony taken from the Merriam-Webster dictionary ("use of words to express something other than and especially the opposite of the literal meaning"). We decided to suggest them a guiding definition for two reasons. First, hearers do not usually focus on literal vs. non literal meaning, as shown by studies measuring processing times for both types of statements (Inhoff et al., 1984). Therefore, when asked to rephrase the speakers' intended meaning, hearers would have probably come up with sentences expressing the speaker's imagined discursive goals, rather than disclosing their perceived literal meaning. Second, it is reasonable to assume that Turkers would have looked up the standard meaning of ironic utterance given by an online dictionary to ease up their task, possibly coming up with biased definitions.

The Turkers were instructed to consider the entire message in their verbalization to avoid asymmetry in length between the S_{im} and H_{int}. We obtained a dataset of 5,000 S_{im}-H_{int} pairs where five Turkers rephrase each S_{im}. A total of 184 Turkers participated in the rephrasing task. Table 1 shows examples of speaker's ironic messages (S_{im}) and their corresponding hearers' interpretations (H^i_{int}). Next, we ran a second MTurk task to verify whether the generated H_{int} messages are plausible interpretations of the ironic messages. This time we employ three Turkers per task and only Turkers who were not involved in the content generation task were allowed to perform this

[1] https://github.com/debanjanghosh/interpreting_verbal_irony

S_{im}	H^1_{int}	H^2_{int}	H^3_{int}
1. Ed Davey is such a passionate, inspiring speaker	Ed Davey is a boring, uninspiring speaker	Ed Davey is such a dull, monotonous speaker	Ed Davey is not a passionate, inspiring speaker
2. can't believe how much captain America looks like me	I wish I looked like Captain America. I need to lose weights	can't believe how much captain America looks different from me	I don't, but I wish I looked like Captain America
3. Pictures of you holding dead animal carcasses are so flattering	Hate hunting season and the pictures of you holding dead animal are so gross	Pictures of you holding dead animal carcasses is an unflattering look	Pictures of you holding dead animal carcasses are not flattering

Table 1: Examples of speaker's ironic messages (S_{im}) and interpretations given by 3 Turkers (H^i_{int}).

task. We observe that Turkers labeled 5% (i.e., 238 verbalizations) of H_{int}s as invalid and low quality (e.g., wrong interpretation). For both tasks, we allowed only qualified Turkers (i.e., at least 95% approval rate and 5,000 approved HITs), paid 7 cents/task and gave sixty minutes to complete each task. The final dataset contains 4,762 pairs S_{im}-H_{int}.

3 Semantic Incongruity in Ironic Messages: Explicit vs. Implicit

Attardo (2000) and later Burgers (2010) distinguish between two theoretical aspects of irony: *irony markers* and *irony factors*. Irony markers are meta-communicative signals, such as interjections or emoticons that alert the reader that an utterance might be ironic. In contrast, irony factors cannot be removed without destroying the irony, such as the incongruity between the literal evaluation and its context ("semantic incongruity"). Incongruity expresses the contrast between the conveyed *sentiment* (usually, positive) and the targeted *situation* (usually, negative). This contrast can be explicitly or implicitly expressed in the ironic message.

Following Karoui et al. (2017), we consider that semantic incongruity is explicit, when it is lexicalized in the utterance itself (e.g., both the positive sentiment word(s) and the negative situation are available to the reader explicitly). On Twitter, beside sentiment words, users often make use of hashtags (e.g., "Studying 5 subjects …#worstsaturdaynight") or an image (e.g., "Encouraging how Police feel they're above the law. URL"; the URL shows a police car not paying parking) to express their sentiment. We consider these cases as explicit, since the incongruity is present in the utterance even if via hashtags or other media. For implicit incongruity, we consider cases where one of the two incongruent terms ("propositions" in Karoui et al. (2017)) is not lexicalized and has to be reconstructed from the context (either outside word knowledge or a larger conversational context). For example "You are such a nice friend!!!", or "Driving in Detroit is fun ;)" are cases of ironic messages where the semantic incongruity is implicit. Based on these definitions of explicit and implicit incongruity, two expert annotators annotated the S_{im}-H_{int} dataset (1000 ironic messages) as containing explicit or implicit semantic incongruity. The inter-annotator agreement was κ=0.7, which denotes good agreement similar to Karoui et al. (2017). The annotation showed that 38.7% of the ironic messages are explicit, while 61.3% are implicit. In the following section we propose a typology of linguistic strategies used in hearers' interpretations of speakers' ironic messages and in section 6.2 we discuss the correlation of linguistic strategies with the type of semantic incongruity.

4 Interpreting Verbal Irony: A Typology of Linguistic Strategies

Given the definition of verbal irony, we would expect that Turkers' interpretation of speaker's ironic message will contain some degree of opposite meaning with respect to what the speaker has said. However, it is unclear what linguistic strategies the Turkers will use to express that. To build our typology, from the total set of S_{im}-H_{int} pairs obtained through crowdsourcing (i.e., 4,762 pairs; see Section 2) we selected a *dev* set of 500 S_{im}-H_{int} pairs. Our approach does not assume any specific theory or irony, but it is data-driven: a linguist expert in semantics and pragmatics analyzed the *dev* set to formulate the lexical and pragmatic phenomena attested in the data. The assembled typology is, thus, the result of a bottom-up procedure. A S_{im}-H_{int} pair can be annotated with more than one strategy. The core linguistic strategies are explained below and synthesized in Table 2.

Typology	Distribution (%)
Antonyms	
- lexical antonyms	(42.2)
- antonym phrases	(6.0)
Negation	
- simple negation	(28.4)
Antonyms OR Negation	
- weakening sentiment	(23.2)
- interrogative $\rightarrow$ declarative	(5.2)
- desiderative constructions	(2.8)
Pragmatic inference	(3.2)

Table 2: Typology of linguistic strategies and their distribution (in %) over the *dev* set

4.1 Linguistic Strategies

Lexical and phrasal antonyms: This category contains lexical antonyms (e.g., "love" $\leftrightarrow$ "hate", "great" $\leftrightarrow$ "terrible") as well as indirect antonyms (Fellbaum, 1998), where the opposite meaning can only be interpreted in context (e.g., "passionate speaker" $\rightarrow$ "boring speaker"; Table 1). Although the typical antonym of "passionate" is "unpassionate", "boring" works in this context as a lexical *opposite* since a speaker who is passionate entails that he is not boring. Besides lexical antonyms, Turkers sometimes use antonym phrases (e.g., "I can't wait" $\rightarrow$ "not looking forward", "I like (to visit ER)" $\rightarrow$ "I am upset (to visit ER) ").

Negation: Here, Turkers negate the main predicate. This strategy is used in the presence of copulative constructions where the predicative expression is an adjective/noun expressing sentiment (e.g., "is great" $\rightarrow$ "is **not** great") and of verbs expressing sentiment (e.g., "love" $\rightarrow$ "**do not** love") or propositional attitudes (e.g., "I wonder" $\rightarrow$ "I **don't** wonder").

Weakening the intensity of sentiment: The use of negation and antonyms is sometimes accompanied by two strategies that reflect a weakening of sentiment intensity. First, when S_{im} contains words expressing a high degree of positive sentiment, the hearer's interpretation replaces them with more neutral ones (e.g., "I **love** it" $\rightarrow$ "I don't **like** it"). Second, when S_{im} contains an intensifier, it is eliminated in the Turkers' interpretation. Intensifiers specify the degree of value/quality expressed by the words they modify (Méndez-Naya, 2008) (e.g., "cake for breakfast. **so** healthy" $\rightarrow$ "cake for breakfast. **not** healthy").

Interrogative to Declarative Transformation (+ Antonym/Negation): This strategy, used mostly in conjunction with the negation or antonym strategies, consists in replacing the interrogative form with a declarative form, when S_{im} is a rhetorical question (for brevity, RQ) (e.g., "don't you **love** fighting?" $\rightarrow$ "I **hate** fighting").

Counterfactual Desiderative Constructions: When the ironic utterance expresses a positive/negative sentiment towards a past event (e.g., "glad you relayed this news") or an expressive speech act (e.g., "thanks X that picture needed more copy") the hearer's interpretation of intended meaning is expressed through the counterfactual desiderative constructions *I wish (that) p* ("**I wish** you hadn't relayed ...", "**I wish** X didn't copy ..."). Differently from antonymic phrases, this strategy stresses on the failure of the speaker's expectation more than on their commitment to the opposite meaning.

Pragmatic Inference: In addition to the above strategies, there are cases where the interpretation calls for an inferential process to be recognized. For instance, "made 174 this month ... *I'm gonna buy a yacht!*" $\rightarrow$ "made 174 this month ... **I am so poor**". The distribution of the strategies on the *dev* set is represented in Table 2.

4.2 Links to Theoretical Frameworks

In linguistic literature many different approaches to irony have been provided. Here we focus on the three accounts (w.r.t. examples from S_{im}-H_{int} corpus) that bear a different views on pragmatic factors. According to Grice (1975), ironic messages are uttered to convey a meaning opposite to that literally expressed, flouting the conversational maxim of quality "do not say what you believe to be false". In verbal irony, the violation of the maxim is frequently signaled by "the opposite" of what is said literally (e.g., intended meaning of "carcasses are flattering" is they are gross; Table 1). The linguistic strategies of *antonyms* (e.g. "worst day of my life") and simple *negation* ("yeap we totally dont drink alcohol every single day"[...]) cover the majority of the S_{im}-H_{int} corpus and seem to fit the Gricean (Grice, 1975) account of irony, since the hearer seems to have primarily recognized the presence of semantic incongruity. However, as touched upon by Giora (1995), *antonyms* and *direct negation* are not always semantically equivalent strategies, since the second sometimes allows a graded interpretation: if "x is not encouraging", it is not necessarily bad, but simply “x < encouraging”. Such an implicature is available exclusively with items allowing mediated contraries, such as sentiment words (Horn, 1989). Direct negation with sentiment words implies that just one value in a set is negated, while the others are potentially affirmed. The spectrum of interpretations allowed by negation as a rephrasing strategy indicates that hearers recognize that the *relevance* of the ironic utterance in itself plays a role next to what the utterances refers to (if the rephrased utterance is intended as “x is not encouraging at all”, the perceived irrelevance of the corresponding ironic utterance is more prominent than in “x is not very encouraging”). The fact that the interpretation of irony has a propositional scope is even clearer when the ironic sentence in interrogative form (“and they all lived happily ever after ?”) is rephrased as a declarative (e.g. “I doubt they all lived happily ever after”): the hearers recognizes that the question has a rhetoric value since otherwise contextually irrelevant. The intentional falsehood of Gricean analysis is also not deemed by Sperber and Wilson (1986); Wilson and Sperber (2012) as a necessary and sufficient condition for irony. According to their theory of *echoic mentioning*, irony presupposes the mention to the inappropriateness of the entire sentence: in asserting “awesome weather in Scotland today” the speaker does not simply want to express that the weather was horrible but he signals that assuming that the weather would be nice was irrelevant and, thus, ridiculous. Kreuz and Glucksberg (1989) expand the Relevance Theory approach talking about *echoic reminding* to account for cases such as “could you be just a little louder, please? My baby isn't trying to sleep” where the extreme politeness reminds the hearer that the question is indeed a request and that the mother bears a certain stance and has certain expectations towards the addressee. Similarly, the use of the *pragmatic inference* strategy cannot be fully explained in Gricean terms: the rephrase “made 174 this month . . . I am so poor” for “made 174 this month . . . I am gonna buy a yatch” more than pointing to the presence of lexical incongruity, show that the hearers knows for background knowledge that the assertion of “buying a yatch” is completely irrelevant in the context of a low salary situation. Rephrasing strategies using counterfactual desiderative constructions (e.g. “I really wish my friends and family would check up on my after yesterday's near death experience”) show, instead, that the interpretation of irony involves an *echoic reminding* to the speaker's (social) expectations which failed to be fulfilled. Overall, using the results of our crowdsourcing experiment with main existing theories of irony, it turns out that the theories have a complementary explanatory power. In Section 6.2 we investigate weather this situation might relate to the presence of explicit/implicit irony.

5 Empirical Analysis of Interpretation Strategies

Here our goal is to perform a comparative empirical analysis to understand how hearers interpret verbal irony. To accomplish this, we propose computational models to automatically detect these linguistic strategies in two datasets: (1) S_{im} -H_{int} dataset and (2) the $SIGN$ dataset. As stated in Section 2, albeit for a different purpose, the task designed in Peled and Reichart (2017) is identical to ours: they used a set of 3,000 sarcastic tweets and collected five interpretation verbalization, including an option to just copy the original message if it was not deemed ironic. They used workers skilled in comedy writing and literature paraphrasing. $SIGN$ contains 14,970 pairs. To evaluate our models, we asked two annotators to annotate two $test$ sets of 500 pairs each from the S_{im} -H_{int} and the $SIGN$ dataset (i.e., denoted by $SIGN_{test}$), respectively. Note, the $test$ set for the S_{im} -H_{int} has no overlap with the dev set of 500 S_{im}-H_{int} pairs used to identify the strategies (Section 4). Agreement between the annotators for both sets is high with $\kappa > 0.9$. In $SIGN_{test}$, 79 instances were just copies of the original message, which we eliminated, thus the $SIGN_{test}$ contains only 421 instances.

5.1 Computational Methods

Lexical Antonyms. To detect whether an S_{im}-H_{int} pair uses the *lexical antonyms* strategy, we first need to build a resource of lexical antonyms. We use the MPQA sentiment Lexicon (Wilson et al., 2005), Hu and Liu (2004)'s opinion lexicon, antonym pairs from Mohammad et al. (2013), antonyms from WordNet, and pairs of opposite verbs from Verbocean (Chklovski and Pantel, 2004).

Given this lexicon of lexical antonyms, the task is now to detect whether a given S_{im}-H_{int} pair

	dev			*test*			$SIGN_{test}$		
Strategies	P	R	F1	P	R	F1	P	R	F1
Lex_ant	89.0	95.7	92.2	97.2	89.9	93.4	89.4	97.9	93.5
Simple_neg	92.0	89.4	90.7	88.3	88.3	88.3	93.3	91.2	92.2
AN_weaksent	93.6	87.9	90.7	95.0	91.9	93.4	93.3	87.5	90.3
$AN_{I \rightarrow D}$	53.1	65.4	58.6	80.0	0.44	57.2	85.7	70.6	77.4
AN_desiderative	100.0	92.9	96.3	100.0	100.0	100.0	100.0	66.7	80.0
AntPhrase+PragInf	86.2	53.2	65.8	70.7	85.3	77.4	89.5	68.0	77.3

Table 3: Evaluation of Computational Methods on dev, $test$ and $SIGN_{test}$ set (in %)

uses the *lexical antonyms* strategy. We use a heuristic approach based on word-alignment and dependency parsing (similar to contradiction detection (De Marneffe et al., 2008)). Word-to-word alignments between S_{im}-H_{int} are extracted using a statistical machine translation (SMT) alignment method - IBM Model 4 with HMM alignment from Giza++ (Och and Ney, 2004). We consider a lexical antonym strategy if: 1) antonym words are aligned; 2) they are the roots of the respective dependency trees or if the nodes modified by the lexical antonyms are the same in their respective trees (e.g., 'can you show any **more** of steelers" → "show **less** of steelers", the candidate lexical antonyms are *more* and *less* and they are the objects of the same predicate in S_{im}-H_{int}: **show**). Out of 211 S_{im}-H_{int} pairs that are marked as having *lexical antonym* strategy (dev set), 12 instances are identified by only the dependency parses, 67 instances by the word-alignments, and 100 instances by both (P/R/F1 scores are 92.1%, 77.7% and 84.3%), respectively on dev dataset. However, sometimes both dependency and word-alignment methods fail. In "circling down the bowl. **Yay**" → "circling down the bowl. **awful**", although the lexical antonyms **yay** and **awful** exist, neither the alignment nor the dependency trees can detect it (25 such instances in the dev set). To account for this, after having run the dependency and alignment methods, we also just search whether a S_{im}-H_{int} pair contains a lexical antonym pair. This improves the final recall and on the dev set we achieve 89.0% precision, 95.7% recall, and 92.2% F1 on dev dataset (Lex_ant Strategy; Table 3 show results both on dev and the $test$ sets). Note, just searching whether a lexical antonym pair is present in a S_{im}-H_{int} pair results in low precision (58.6%) but high recall (80%).

Simple negation. This strategy (denoted as Simple_neg in Table 3 and Table 4) involves identifying the presence of negation and its scope. Here, however, the scope of negation is constrained since generally Turkers negated only a single word (i.e., "love" → "**not** love"). Thus our problem is easier than the general problem of finding the scope of negation (Li and Lu, 2018; Qian et al., 2016; Fancellu et al., 2016). We use 30 negation markers from Reitan et al. (2015) to find negation scope in tweets. We first detect whether a negation marker appears in either H_{int} or S_{im}, but not in both (negation can appear in S_{im} for ironic blame) If the marker is used, we extract its parent node from the dependency tree, and if this node is also present in the other utterance, then *Negation* strategy is selected. For instance, in "**looks** just like me" → "does **not look** like me", the negation **not** is modifying the main predicate **looks** in H_{int}, which is also the main predicate in S_{im} (words are lemmatized). In the next section, we discuss if the parent nodes are not the same but similar and with different sentiment strength.

Weakening the intensity of sentiment. The first strategy — replacing words expressing a high degree of positive/negative sentiment with more neutral ones ('I **love** being sick" → "I **don't like** being sick)—, is applied only in conjunction with the negation strategy. We measure the difference in strength using the Dictionary of Affect (Whissell et al., 1986). Out of 31 S_{im}-H_{int} pairs in the dev set, we automatically identify 28 interpretations that use this approach. For the second strategy — removing the intensifier (I am **really** happy" → "I am disappointed') —, we first determine whether the intensifier exists in S_{im} and is eliminated from H_{int}. We use only adjective and adverb intensifiers from Taboada et al. (2011), primarily to discard conjunctions such as "so" ("no water **so** I can't wash ..."). This strategy is used together with both *lexical antonyms* and *Simple negation* strategies. For a candidate S_{im}-H_{int} pair, if the *lexical antonym* strategy is selected and a_S and a_H are the lexical antonyms, we determine whether any intensifier modifies a_S and no intensifier modifies a_H. If the *Negation* strategy is selected, we identify the negated term in the H_{int} and then search its aligned node from the S_{im} using the word-word alignment. Next, we search in the S_{im} if any intensifier is intensifying the aligned term. The strategies are denoted as AN_weaksent in Table 3 and Table 4.

Interrogative to Declarative Transformation (+ Antonym/Neg). To capture this strategy we need to determine first if the verbal irony was expressed as a rhetorical question. To build a classifier to detect RQ, we collect two categories of tweets (4K each) (1) tweets labeled with #sarcasm or #irony that also contain "?", and (2) information seeking tweets containing "?". We train a binary classifier using SVM RBF Kernel with default parameters. The features are Twitter-trained word embeddings (Ghosh et al., 2015), modal verbs, pronouns, interrogative words, negations, and position of "?" in a tweet. We evaluate the training model on the dev data and the P/R/F1 are 53.2%, 65.4%, and 58.6%, respectively (in future work we plan to develop more accurate models for RQ detection). Once we detect the ironic message was expressed as a RQ, we identify the specific interpretation strategy accompanying the transformation from interrogative to declarative form: antonym or negation. These combined strategies are denoted as $\mathrm{AN}_{I\to D}$ in Table 3 and Table 4.

Desiderative Constructions: Currently, we use a simple regular expression "I [w]∗ wish" to capture counterfactual cases (AN_desiderative in Tables 3 and Table 4).

Note, when the *Simple negation* and *lexical antonyms* strategies are combined with other strategy (e.g., removing of intensifier), we consider this combined strategy for the interpretation of verbal irony and not the *simple negation* or *lexical antonym* strategy (i.e., we do not double count).

Phrasal antonyms and pragmatic inference: Identifying phrasal antonyms and pragmatic inference is a complex task, and thus we propose a method of phrase matching based on phrase extraction via unsupervised alignment technique in SMT. We use IBM Model 4 with HMM (Giza++; (Och and Ney, 2000)), phrase extraction via Moses (Koehn et al., 2007) and the IRST tool to build the required language models. As post-processing, we first remove phrase pairs obtained from the S_{im}-H_{int} bitext that are also present in the set of extracted phrases from the H_{int}-H_{int} bitext. This increases the likelihood of retaining semantically opposite phrases, since phrases extracted from the H_{int}-H_{int} bitext are more likely to be paraphrastic. Second, based on the translation probability scores ϕ, for phrase *e* if we have a set of aligned phrases f_{set} we reject phrases that have ϕ scores less than $\frac{1}{size(f_{set})}$. Finally, 11,200 phrases are extracted from the S_{im}-H_{int} bitext. The low recall for this strategy is expected since there are too many ways that users can employ pragmatic inference or rephrase the utterance without directly using any antonym or negation. In future, we will explore neural MT (Cho et al., 2014) and use external data to generate more phrases. Since we have not manually evaluated these phrase pairs, we only use this strategy after we have tried all the remaining strategies (AntPhrase+PragInf in Table 3 and Table 4).

Strategies	S_{im}-H_{int}	$SIGN$
Lex_ant	2,198 (40.0)	9,691 (51.8)
Simple_neg	1,596 (29.1)	3,827 (20.5)
AN_weaksent	895 (16.3)	2,160 (11.6)
$\mathrm{AN}_{I\to D}$	329 (6.0)	933 (5.0)
AN_desiderative	92 (1.7)	86 (0.5)
AntPhrase+PragInf	357 (6.5)	1912 (10.1)

Table 4: Distribution of interpretation strategies on two datasets (in %)

5.2 Results and Distribution of Linguistic Strategies

The performance of the models is similar on both $test$ and $SIGN_{test}$ sets, showing consistently good performance (Table 3; 90% F1 for all strategies, except the AntPhrase+PragInf and $\mathrm{AN}_{I\to D}$). Given these results, we can now apply these models to study the distribution of these strategies in the entire datasets (Table 4). The strategy distribution between our dataset S_{im}-H_{int} and $SIGN$ dataset is similar and matches the distribution on the manual annotations on the dev dataset in Table 2. The sum of the strategies can exceed the total number of the pairs since a tweet can contain several ironic sentences that are interpreted by Turkers. For instance, in "Dave too **nice** . . . a **nice** fella" → "Dave not nice . . . a mean fella" we observe the application of two strategies, *lexical antonyms* (e.g., **nice** → **mean**) and *negation* (e.g., **nice** → **not nice**).

6 Discussion

6.1 Hearer-dependent Interpretation Strategies

We investigate how hearers adopt strategies for interpreting the speaker's ironic intent. To implement this study, we selected three Turkers (e.g., H^1, H^2, and H^3; In Table 1, H^i_{int} are generated by the correspondent Turker H^i), from our crowd-sourced data, who were able to rephrase at least five hundred identical S_{im} messages. Note, we cannot carry this experiment on the $SIGN$ dataset (Peled and Reichart, 2017) because the annotators' information is absent there.

Although the three Turkers choose *lexical antonym* and *simple negation* as two top choices, there is some variation among them. H^1 and H^2 choose *antonyms* more frequently than *negation* while in contrary Turker H^3 choose *negation* more than *antonyms*, sometime combined with the *weakening of sentiment* strategy. As we mentioned in Section 4.2, antonyms and direct negation are not semantically equivalent strategies since the latter, allows a graded interpretation: if "x is not inspiring", it is not necessarily bad, but simply "x $<$ inspiring" (Giora, 1995). In Table 1, the S_{im}-H_{int} pair "passionate" $\rightarrow$ "boring" and "flattering" $\rightarrow$ "gross" (interpretation of H^1) have more contrast than the pair "passionate" $\rightarrow$ "not passionate" and "so flattering" $\rightarrow$ "not flattering" (interpretation of H^3). This suggests that H^1 perceive the intensity of negative sentiment towards the target of irony ("Ed Davey" and "picture of dead animals", respectively) higher than Turker H^3. All three Turkers have chosen the remaining strategies with similar frequencies.

6.2 Message-dependent Interpretation Strategies

Interpretation Strategies and the Type of Semantic Incongruity: We investigate whether the type of semantic incongruity in the ironic message (explicit vs. implicit; see Section 3) influences the choice of interpretation strategies by the hearers. To do this, we looked at S_{im}-level distribution of interpretation strategies used by the hearers for the same ironic message S_{im}. Table 5 represents the correlation of linguistic strategies with the type of semantic incongruity (explicit vs. implicit) as well as the presence and absence of irony markers.

We notice that Turkers use lexical antonyms as interpretation strategy more when the semantic incongruity is explicit than implicit (48.5% vs. 34.8%): the presence of explicit sentiment triggered the use of the antonym strategy. In contrary they use simple negation more when the semantic incongruity is implicit than explicit.

Strategies	*incongruity*		*marker*	
	Exp.	*Imp.*	+	−
Lex_ant	*48.5*	34.8	35.7	*42.2*
Simple_neg	24.9	32.3	28.9	30.0
AN_weaksent	14.3	17.6	15.7	16.8
$AN_{I \rightarrow D}$	5.9	6.1	12.3	3.1
AN_desiderative	1.3	1.9	0.9	2.0
AntPhrase+PragInf	5.2	7.1	6.2	6.6

Table 5: Rephrasing Strategies against Incongruency and Irony Markers on S_{im}-H_{int} dataset (in %)

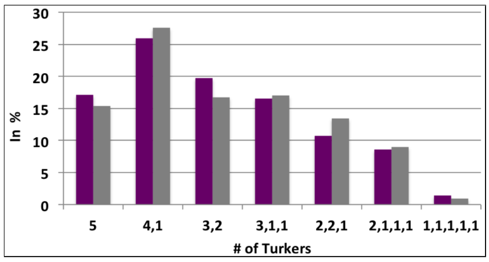


Figure 1: Strategies selected per message (in %)

We also analyze the interpretation strategies w.r.t. to the presence (+) or absence (−) of irony markers. We implement various morpho-syntactic as well as typographic markers (similar to (Ghosh and Muresan, 2018)) to identify the presence of markers. We observe that Lex_ant strategy is used more in cases where the markers are absent. In S_{im}-H_{int}, markers are present twice as much in the case of implicit (21%) than explicit incongruity (10%). This finding validates (Burgers et al., 2012) who argued speakers will likely use markers to signal their ironic intent in implicit incongruity.

Message interpreted the same by all hearers: In Figure 1, the vertical columns (purple: S_{im}-H_{int} and grey: $SIGN$) depict the distribution (in %) of tweets strategy-wise. In S_{im}-H_{int} dataset, for 17% of messages (124 S_{im}s) all five Turkers use the same strategy to interpret the S_{im}s (labeled as *5* on the X-axis), whereas for 26% (188 S_{im}s), 4 Turkers used same strategy (labeled as 4,1 on X-axis) and so on.

We observe when the S_{im}s are marked by strong subjective words e.g., "great", "best", etc., they have been replaced in 90% of cases as lexical antonyms (e.g., "great" → "terrible"). In addition, the majority of adjectives are used in attributive position (i.e., "**lovely** neighbor is vacuuming at night"), thus blocking paraphrases involving predicate negation. However, not all strong subjective words guarantee the use of direct opposites in the H_{int}s (e.g., "flattering" → "not flattering"; See Table 1). The choice of strategies may also depend upon the target of ironic situation (Ivanko and Pexman, 2003). We implement the bootstrapping algorithm from Riloff et al. (2013) to identify ironic situations in S_{im}s that are rephrased by *Lexical antonym* strategy. We find utterances containing stereotypical negative situations regarding *health issues* (e.g., "having migraines", "getting killed by chemicals") and other undesirable negative states such as "oversleeping", "luggage lost", "stress in life" are almost always interpreted via *lexical antonym* strategy.

Utterances where all five Turkers used *simple negation*, if negative particles are positioned in the ironic message with a sentential scope (e.g., "not a biggie", "not awkward") then they are simply omitted in the interpretations. This trend can be explained according to the inter-subjective account of negation types (Verhagen, 2005). Sentential negation leads the addressee to open up an alternative mental space where an opposite predication is at stake.

7 Related Work

Most NLP research on verbal irony or sarcasm has focused on the task of *sarcasm detection* treating it as a binary classification task using either the utterance in isolation or adding contextual information such as conversation context, author context, visual context, or cognitive features (González-Ibáñez et al., 2011; Liebrecht et al., 2013; Wallace et al., 2014; Zhang et al., 2016; Ghosh and Veale, 2016; Schifanella et al., 2016; Xiong et al., 2019; Castro et al., 2019). Unlike this line of work, our research focuses on how the hearer *interprets* an ironic message. The findings from our study could have multiple impacts on the sarcasm detection task. First, interpretation strategies open up a scope of "graded interpretation" of irony instead of only a binary decision (i.e., predicting the **strength** of irony). Second, nature of semantic incongruence and stereotype irony situations can be useful features in irony detection.

Recently, Peled and Reichart (2017) proposed a computational model based on SMT to generate interpretations of sarcastic messages. We aim to deepen our understanding of such interpretations by introducing a typology of linguistic strategies. We study the distribution of these strategies via both hearer-dependent and message-dependent interpretations. Psycholinguistics studies that have dealt with the hearers' perception, have mainly focused on how ironic messages are processed: through the analysis of reaction times (Gibbs, 1986; Katz et al., 2004), the role of situational context (Ivanko and Pexman, 2003) and in tackling speaker-hearer social relations by annotating ironic texts from different genres (Burgers, 2010). However, no attention has been paid to correlations between how ironic message is expressed and how it is interpreted by the hearer, including what linguistic strategies the hearers employ.

8 Conclusions

We leveraged a crowdsourcing task to obtain a dataset of ironic utterances paired with the hearer's verbalization of their interpretation. We proposed a typology of linguistic strategies for verbal irony interpretation and designed computational models to capture these strategies with good performance. Our study shows (1) Turkers mostly adopt lexical antonym and negation strategies to interpret speaker's irony, (2) interpretations are correlated to stereotype ironic situations, and (3) irony expression (explicit vs. implicit incongruity and absence or presence of markers) influences the choice of interpretation strategies and match with different explanatory theories (the Gricean approach links up better with explicit incongruity, while *Relevance Theory* with the implicit one). The latter can have an impact on irony detection by bringing out more discriminative semantic and pragmatic features.

Acknowledgements

We thank Rituparna Mukherjee, Daniel Chaparro, Pedro Pérez Sánchez, and Renato Augusto Vieira Nishimori who helped us in annotating as well as in running experiments. This paper partially based on the work supported by the DARPA-DEFT program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976* .

Salvatore Attardo. 2000. Irony markers and functions: Towards a goal-oriented theory of irony and its processing. *Rask* 12(1):3–20.

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Christian Burgers, Margot Van Mulken, and Peter Schellens. 2012. Verbal irony differences in usage across written genres. *Journal of Language and Social Psychology* 31(3):290–310.

Christian Frederik Burgers. 2010. *Verbal irony: Use and effects in written discourse*. [Sl: sn].

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _obviously_ perfect paper). *arXiv preprint arXiv:1906.01815* .

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*. volume 4, pages 33–40.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. CoNLL '10.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *ACL*. volume 8, pages 1039–1047.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. volume 1, pages 495–504.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1615–1625. https://doi.org/10.18653/v1/D17-1169.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT*. pages 161–169.

Aniruddha Ghosh and Tony Veale. 2017. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 482–491. https://doi.org/10.18653/v1/D17-1050.

Debanjan Ghosh, Alexander R. Fabbri, and Smaranda Muresan. 2018. Sarcasm analysis using conversation context. *Computational Linguistics* 44(4):755–792. https://doi.org/10.1162/coli_a_00336.

Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1003–1012. http://aclweb.org/anthology/D15-1116.

Debanjan Ghosh and Smaranda Muresan. 2018. " with 1 follower i must be awesome: P." exploring the role of irony markers in irony recognition. In *Twelfth International AAAI Conference on Web and Social Media*.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Saarbrcken, Germany, pages 186–196. http://aclweb.org/anthology/W17-5523.

Raymond W Gibbs. 1986. On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General* 115(1):3.

Rachel Giora. 1995. On irony and negation. *Discourse processes* 19(2):239–264.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*. Association for Computational Linguistics, pages 581–586.

H Paul Grice. 1975. " logic and conversation" in cole, p., and morgan, j.(eds.). *Syntax & Semantics* 3.

Henk Haverkate. 1990. A speech act analysis of irony. *Journal of Pragmatics* 14(1):77–109.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. Cascade: Contextual sarcasm detection in online discussion forums. In *Proceedings of* the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, pages 1837–1848. http://aclweb.org/anthology/C18-1156.

Laurence Horn. 1989. *A natural history of negation*. The University of Chicago Press.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 168–177.

Albrecht Werner Inhoff, Susan D Lima, and Patrick J Carroll. 1984. Contextual effects on metaphor comprehension in reading. *Memory & Cognition* 12(6):558–567.

Stacey L Ivanko and Penny M Pexman. 2003. Context incongruity and irony processing. *Discourse Processes* 35(3):241–279.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *ACL (2)*. pages 757–762.

Jihen Karoui, Farah Benamara, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. Association for Computational Linguistics (ACL).

Albert N Katz, Dawn G Blasko, and Victoria A Kazmerski. 2004. Saying what you don't mean: Social influences on sarcastic language processing. *Current Directions in Psychological Science* 13(5):186–189.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, pages 177–180.

Roger J Kreuz. 2000. The production and processing of verbal irony. *Metaphor and Symbol* 15(1-2):99–107.

Roger J Kreuz and Sam Glucksberg. 1989. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of experimental psychology: General* 118(4):374.

Hao Li and Wei Lu. 2018. Learning with structured representations for negation scope extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 533–539.

CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Diana Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of LREC*.

Belén Méndez-Naya. 2008. Special issue on english intensifiers. *English Language and Linguistics* 12(02):213–219.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1095–1104. https://doi.org/10.18653/v1/P16-1104.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics* 39(3):555–590.

Smaranda Muresan, Roberto Gonzalez-Ibanez, Debanjan Ghosh, and Nina Wacholder. 2016. Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology* http://dx.doi.org/10.1002/asi.23624.

Franz Josef Och and Hermann Ney. 2000. Giza++: Training of statistical translation models.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational linguistics* 30(4):417–449.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pages 2854–2859.

Lotem Peled and Roi Reichart. 2017. Sarcasm sign: Interpreting sarcasm with sentiment based monolingual machine translation. *arXiv preprint arXiv:1704.06836* .

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and Wei Luo. 2016. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 815–825.

Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 99–108.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 704–714.

Rossano Schifanella, Paloma de Juan, Joel Tetreault, and Liangliang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, pages 1136–1145.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and cognition*, volume 142. Harvard University Press Cambridge, MA.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1010–1020. http://aclweb.org/anthology/P18-1093.

Arie Verhagen. 2005. *Constructions of intersubjectivity: Discourse, syntax, and cognition*. Oxford University Press on Demand.

Byron C Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *ACL (2)*. pages 512–516.

Cynthia Whissell, Michael Fournier, Rene Pelland, Deborah Weir, and Katherine Makarec. 1986. A dictionary of affect in language: Iv. reliability, validity, and applications. *Perceptual and Motor Skills* 62(3):875–888.

Deirdre Wilson and Dan Sperber. 2012. Explaining irony. *Meaning and relevance* pages 123–145.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The World Wide Web Conference*. ACM, pages 2115–2124.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *COLING*.

Inflectional networks: Graph-theoretic tools for inflectional typology

Andrea D. Sims
The Ohio State University
sims.120@osu.edu

Abstract

The interpredictability of the inflected forms of lexemes is increasingly important to questions of morphological complexity and typology, but tools to quantify and visualize this aspect of inflectional organization are lacking, inhibiting effective cross-linguistic comparison. In this paper I use metrics from graph theory to describe and compare the organizational structure of inflectional systems. Graph theory offers a well-established toolbox for describing the properties of networks, making it ideal for this purpose. Comparison of nine languages reveals previously unobserved generalizations about the typological space of morphological systems. This is the first paper to apply graph-theoretic tools to the goal of inflectional typology.

1 Introduction

Morphological typology has long classified languages in terms of how words are built out of morphemes. A typical formulation defines three or four types: isolating, agglutinative, fusional, and sometimes polysynthetic. More nuanced work seeks to break the types down into their component properties, with languages compared based on clusters of these (Plank, 1999). This newer approach is better able to capture cross-linguistic diversity, but it gives priority to the same aspects of morphological structure as the traditional classification scheme: syntagmatic relationships between formal elements (e.g. how many morphemes there are per word, known as the degree of synthesis (Comrie, 1981)), and the extent to which form-meaning mappings are isomorphic (e.g. as opposed to the language having inflection classes).

Morphological typologies built on these priorities fail to capture important aspects of morphological structure, corresponding to a distinction between two broad notions of morphological complexity that Ackerman and Malouf (2013) call Enumerative Complexity (E-complexity) and Integrative Complexity (I-complexity). E-complexity has to do with the size of a morphological system, e.g., the number of cells in lexemes' paradigms, the system's degree of synthesis, or the number of its inflection classes. I-complexity, on the other hand, has to do with the predictability of the inflected forms of lexemes. A morphological system is I-complex to the extent that the inflected forms of a newly encountered lexeme are unpredictable. This is a function of the distribution of elements in the system. Even systems with high E-complexity, such as a large number of inflection classes, may have low I-complexity, if morphological elements are distributed in ways that make them predictable (Ackerman and Malouf, 2013; Cotterell et al., to appear; Wurzel, 1989). I-complexity is thus oriented to the *internal organization* of inflectional systems, rather than their size. However, this organization is not captured by traditional typological measures.

In this paper I adopt metrics from graph theory, using them to describe and compare the internal organization of inflectional systems.[1] I analyze inflection classes as nodes in a network that are connected by the morphological structure that they have in common; two classes are connected if they use same exponent(s) to realize a set of morphosyntactic values. Conceptualized in this way, inflectional networks reflect the distribution of exponents in a language's inflectional system, and by extension, the internal organization of that system. Graph theory offers an established, widely applied toolkit for describing the properties of networks, making it a natural choice for application. While some interesting and previously unobserved generalizations emerge from comparison of different languages' inflectional networks, the primary goal of this paper is to demonstrate the usefulness of

[1] Data and code are available at https://github.com/sims120/inflectional-networks.

	STOL	MESTO	KNIGA	KOST'
	'table'	'place'	'book'	'bone'
ACC.SG	stol	mesto	knigu	kost'
INS.SG	stolom	mestom	knigoj	kost'ju
DAT.PL	stolam	mestam	knigam	kostjam

Table 1: Partial inflectional paradigms of Russian nouns: three paradigm cells that differ in how informative they are about inflection class membership

applying graph-theoretic tools to inflectional data, and to outline some specific ways to quantify and compare inflectional systems.

Section 2 motivates an approach to typological comparison based on the paradigmatic distribution of exponents within an inflectional system. Section 3 gives a formal definition of an inflectional network. Section 4 discusses methodological choices. Section 5 introduces a variety of standard graph-theoretic measures, illustrating them using Russian noun inflection. Section 6 then compares nine languages' inflectional systems based on a couple of these measures, showing that their organization exhibits cross-linguistic diversity but also notable commonalities. Finally, Section 7 offers some conclusions and future directions.

2 Internal organization as a basis for inflectional typology

Work in the abstractive Word and Paradigm tradition (Blevins, 2006) emphasizes the paradigmatic or 'external' dimension of morphological structure: distributions of inflected word-forms within and across paradigms, and how these give rise to competition among inflectional exponents. In this view, word-internal/syntagmatic structure (e.g. stem-affix relations) is a byproduct of the ways in which words are paradigmatically related within and across inflectional paradigms (Ackerman et al., 2016; Blevins, 2016).

In the inter-paradigmatic direction, a central question has to do with how inflected forms cue inflection class membership – the so-called Paradigm Cell Filling Problem (Ackerman et al., 2009). Table 1 illustrates the issue using a subset of the inflected forms of Russian nouns. (For the moment I assume a typical, four-class description of Russian nouns, although I will ultimately employ a more robust representation in Sections 5 and 6.) In Russian, the accusative singular exponent *-u* (as in *knig-u* 'book-ACC.SG') is fully informative about inflection class membership, which is to say, about what the other forms of the same lexeme are. If a competent adult speaker encounters a neologism ending in *-u* and knows that it is accusative singular, all other forms of the noun are predictable (ignoring stress placement). However, inflected forms are not guaranteed to be fully (or at all) informative in this way. Instrumental singular *-om* is partially informative: the new word must belong to either the STOL class or the MESTO class, but the observed form does not resolve which. The dative plural exponent *-am* is uninformative, since it appears in every inflection class. The distributions of inflected forms across classes thus determine how and the extent to which allomorphs cue inflection class membership. They likewise define a pattern of relatedness among lexemes, and by extension inflection classes, and reflect the internal organization of the inflectional system.

This internal organization has been of particular interest in work that seeks to quantify inflectional complexity. From an I-complexity perspective, the Paradigm Cell Filling Problem is a significant issue because neither child (Lignos and Yang, 2016) nor adult (Bonami and Beniamine, 2016) speech input is sufficient to observe all inflected forms of all lexemes. Speakers must therefore be able to productively predict and generate unobserved inflected forms. The complexity of an inflectional system is a function of the difficulty of this task, given some partial knowledge of a lexeme (Stump and Finkel, 2013).

Estimates of the I-complexity of inflectional systems based on paradigmatic relations – essentially, proportional analogy – have been calculated in set-theoretic (Stump and Finkel, 2013) and information-theoretic terms (Ackerman et al., 2009; Ackerman and Malouf, 2013; Bonami and Beniamine, 2016; Mansfield, 2016; Parker and Sims, to appear; Sims and Parker, 2016; Stump and Finkel, 2013). Sequence-to-sequence neural network models for inflection have also been employed (Cotterell et al., to appear; Malouf, 2017). Using conditional entropy, Parker (2016) estimates the complexity of the Russian nominal system at between 0.5 and 0.6 bits, depending on how much detail about Russian inflectional outcomes is included in the analysis.

This notion of inflectional complexity has also been extended to cross-linguistic comparison. Ackerman and Malouf (2013)[436] propose the Low Entropy Conjecture: "...enumerative morphological complexity is effectively unrestricted, as long as the average conditional entropy, a measure of integrative complexity, is low..." The Low Entropy Conjecture is posited to be a universal constraint on morphological I-complexity, driven by speakers' need to be able to solve the Paradigm Cell Filling Problem. Other work has suggested a trade-off between I-complexity and E-complexity (Cotterell et al., to appear). Importantly, however, both suggest that I-complexity reveals commonalities among languages' inflectional systems that are not captured by typological approaches focused on E-complexity.

As a basis for cross-linguistic comparison, the notion of I-complexity thus reflects something different about morphological structure than traditional measures do. It is also inextricably rooted in the internal organization of inflectional systems – in particular, the distribution of allomorphs across lexemes and classes. Yet tools for directly examining this organization are lacking.[2] Previous work largely boils the distributional properties of an inflectional system down to an estimate of its complexity as a whole (as with Parker's estimate for Russian nouns). While this is appropriate to some goals, single-value measures have the same problem found with all averages: many different distributions can produce the same average. As a basis for comparison across languages this offers an incomplete picture of the extent to which languages are similar or different (Elsner et al., submitted). Moreover, languages seem to differ in the extent to which paradigmatic relations (proportional analogy) are important to maintaining low I-complexity (Sims and Parker, 2016), suggesting the need to directly investigate a system's organization, and not only its resulting complexity.

These issues highlight the need to drill down on the distributional properties of individual morphological elements. Tools are needed for the description of individual systems at that level that offer a basis for meaningful cross-linguistic comparison.

3 Inflectional systems as networks

I define an inflection class system as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set $\mathcal{V}$ of nodes consists of the inflection classes of the language and the set $\mathcal{E}$ of edges consists of unordered pairs of elements in $\mathcal{V}$. In particular, elements in $\mathcal{E}$ are defined by exponence shared among pairs of elements in $\mathcal{V}$. Taking the partial set of inflected forms from Table 1 as a simplified example, there are four inflection classes (thus, $\mathcal{V}(\mathcal{G})$ = {STOL, MESTO, KNIGA, KOST'}). The classes are distinct overall, but all four have the exponent *-am* in dative plural, the classes of STOL and KOST' both lack an overt accusative singular exponent, and STOL and MESTO both have *-om* in instrumental singular. These overlaps define six edges $\mathcal{E}(\mathcal{G})$ = {STOL-MESTO, STOL-KOST', STOL-KNIGA, MESTO-KOST', MESTO-KNIGA, KOST'-KNIGA}, as visualized in Figure 1.[3]

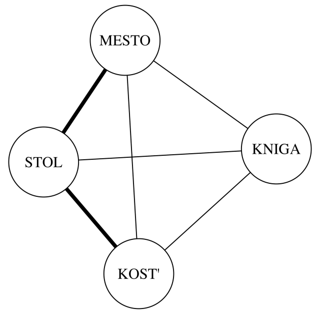


Figure 1: Network graph of the partial set of Russian noun forms shown in Table 1

Furthermore, the weight of an edge is defined as the number of cells in which two classes overlap. This is shown as a heavier line for the edges connecting nodes STOL and MESTO, and STOL and KOST'. Edge weight captures the observation that classes that overlap in more cells are more similar to each other. In language change, these are more likely to analogically influence each other. Edges can thus be thought of as paths of analogical reasoning— more specifically, the edges represent potential pivots for inflection class shift.

4 Segmentation and the definition of classes

The number of inflection classes a given language is analyzed as having is predicated on a segmentation of its words into stems and exponents. Mor-

[2] However, Beniamine (2018) is notable for the use of network visualization.

[3] All network graphs in this paper were plotted with the igraph package (Csardi and Nepusz, 2006) in R (R Core Team, 2019). This package was also used to calculate clustering coefficient, shortest path length, and betweenness centrality, as described in Section 5 below.

phological segmentation has long presented analytic challenges for description and typology (Beniamine et al., 2017a; Hockett, 1947; Nida, 1949), formal theory (Matthews, 1972; Spencer, 2012), and computational modeling (Goldsmith, 2001, 2010; Harris, 1970; Manning, 1998). Encoder-decoder neural models of inflection (Faruqui et al., 2016; Kann and Schütze, 2016; Malouf, 2017; Silfverberg and Hulden, 2018) have recently become popular in part because they are able to sidestep questions of how words should be segmented into morphological units and how to define discrete inflection classes. However, it is difficult to identify and interpret the latent representations that neural network models of inflection actually learn. The analyses below are instead based on manual segmentation, which has the advantage of being maximally linguistically interpretable.[4]

In what follows I use a global segmentation strategy (Beniamine et al., 2017b), in which the 'stem' is the maximal continuous string shared by all inflected forms of a lexeme. There are two exceptions to this principle: 1) Suprasegmental material (e.g. tone) is analyzed separately from segmental material, allowing globally shared segmental material to be identified as part of the stem, even when suprasegmental material is different from one inflected form to another. Suprasegmental material that is not shared by all inflected forms of a lexeme is assigned to the exponent. 2) Purely automatic phonology (e.g. of the type that is vowel harmony in Turkish, or vowel reduction in Russian) is ignored. This method results in bits of form that linguists often classify as stem allomorphy (morphophonological alternations, stem extensions, theme vowels, stress shift, etc.) being assigned to the exponent.[5]

Once a segmentation into stem and exponent is made, defining classes is a trivial matter: two words belong to the same inflection class if and only if the full sets of their exponents are identical. This method results in microclasses in the terminology of Beniamine et al. (2017b), which tend to be large in number, relative to classical descriptions. For example, descriptions of the Russian nominal system tend to posit either three (Vinogradov et al., 1952) or four (Corbett, 1982) (macro)classes, whereas the method used here produces 87 (micro)classes.[6]

Since this is a somewhat unusual analytic choice, it requires some justification. In defining inflection classes, linguists tend to abstract away from morphophonological alternations, especially if phonologically conditioned, preferring to define classes based (solely, ideally) on lexically-conditioned, suppletive exponents. This minimizes the number of inflection classes posited. However, there are at least four reasons to adopt a maximally inclusive definition of exponents, and a more robust number of classes.

First, returning to the Paradigm Cell Filling Problem and the notion of I-complexity, to 'solve' the PCFP speakers must predict entire word-forms. Limiting what counts as an exponent may lead to overestimation or underestimation of the I-complexity of inflectional systems (Elsner et al., submitted; Sims, 2015). This is important because the graph-theoretic approach to inflectional typology argued for in this paper is motivated exactly by a desire to better understand how I-complexity relates to the internal organization of inflectional systems, and the extent of cross-linguistic diversity in this respect.

Second, the line between morphology and phonology cannot always be drawn in a principled and pre-theoretic way. The choice to define exponents in a maximally inclusive way is not theory-neutral, to be sure – it is philosophically aligned with the Word and Paradigm framework. But to the extent that it errs, it does so consistently on the side of representing inflection classes as overly distinct. This is preferable to erring in the opposite direction because we can ask about the extent to which microclasses group into macroclasses, but if we abstract away from morphological differences and thus fail to distinguish two classes in the first place, we will never be able to detect any inter-

[4]A goal for the future is to expand the methods and code to include automatic segmentation of words into stems and exponents, e.g. through integration with the Qumin software package (Beniamine, 2018): https://github.com/XachaB/Qumin

[5]Multiple exponents are treated as a single, combined exponent. To the extent that each of multiple exponents has a separate distribution, an analysis in terms of multilayer networks (Bianchoni, 2018) would likely be needed to capture this. Multilayer network representations are more complex and I leave this extension for the future.

[6]As a reviewer observed, suppletive material is all assigned to the exponent, resulting in maximal differentiation from other classes and potentially increasing not only the number of classes, but the prevalence of disconnected subgraphs. Indeed, exactly this situation is encountered in Russian nouns (see Section 5), showing that segmentation choices affect the representation of the network to some degree. However, it is not clear that there is a 'right' or 'wrong' choice in this respect.

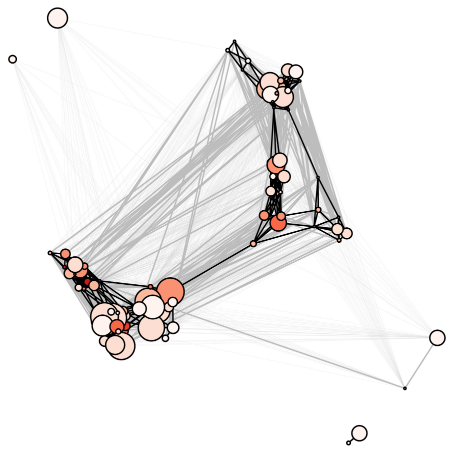


Figure 2: Inflection class system of Russian nouns (87 classes). Nodes size represents the log type frequency of the class. Node color reflects betweenness centrality (darker = more central). Edge color and thickness are according to weight: edges connecting nodes (classes) with the same exponents in more than half of cells are black (N $\geq$ 7); edges connecting nodes with the same exponents in exactly half of cells (N=6) are thick gray; weaker edges are thin gray.

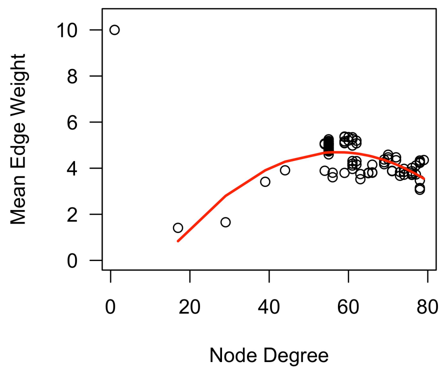


Figure 3: Correlation between node degree and mean edge weight for Russian nouns. The red line shows a quadratic regression fit.

esting aspects of inflectional organization that the abstracted-away-from differences constitute.

Third, as a practical matter, a global segmentation strategy can be applied in a uniform way across languages and requires a minimum of analytic/theoretical assumptions (Beniamine et al., 2017b), evading potential problems created by the use of different analytic methods for different languages.

Finally, and perhaps most importantly, different kinds of allomorphy tend to be found in different types of morphological systems (e.g. agglutinative vs. fusional) (Plank, 1999). Including some kinds of allomorphy and excluding others thus runs the risk of introducing systematic bias into cross-linguistic comparisons of inflection class organization.

In the following section I illustrate how standard measures for network description can be used to quantify the organizational structure of the Russian nominal inflectional system.

5 Network properties of Russian nouns

The inflection class network for Russian nouns is shown in Figure 2. Following Parker (2016), the underlying morphological analysis includes not just regular and productive inflectional suffixes, but also irregular suffixes, stress alternations, stem extensions, defectiveness (no inflected form for a given paradigm cell), and uninflectedness (only one form for all paradigm cells). Node size reflects the log type frequency of the class (i.e. the log number of lexemes it contains), based on 43,486 nouns in Zaliznjak (1977). Node color indicates betweenness centrality, discussed below. Edges are colored according to their weight.

5.1 Number of nodes, edges, and connected components

Basic descriptive statistics for the Russian nominal inflectional network include the number of its nodes ($|\mathcal{V}(\mathcal{G})| = 87$), the number of its edges ($|\mathcal{E}(\mathcal{G})| = 2660$), and how many connected components it has. A connected component is a subgraph containing all of the nodes that are connected via a path. The Russian noun system has two components. One has two nodes that differ from each other only in accusative (the result of animacy-conditioned allomorphy), exemplified by REBËNOK 'child, baby' (NOM.PL *rebjata*), which has a unique suppletive stem alternation *-onOk* $\sim$ *-at*.[7] The remaining 85 classes belong to the other connected component.

5.2 Degree distribution and edge weight

Node degree is the number of edges $\mathcal{K}$ that are connected to a node. In Russian, the large majority of classes have $|\mathcal{K}| > 50$.

[7]Capital O in -onOk indicates a fleeting vowel.

The relationship between node degree and edge weight is shown in Figure 3.[8] The quadratic nature of the distribution (R^2 = 0.55, $p < 0.0001$) probably partly reflects limitations on the extent to which classes can overlap but remain distinct. Classes with both high degree and high edge weight are likely targets for merger, which may explain the relative lack of such classes in Russian nouns. However, interestingly, there is no such restriction for low degree nodes, for which it is entirely possible to overlap with few other classes (low degree), but in many cells (high edge weight). The ways in which Russian nouns overlap thus do not appear to reflect random sampling from the full space of possibilities.[9]

5.3 Clustering coefficient

As is evident visually in Figure 2, Russian inflection classes form clusters: groups of nodes with high-density ties. This clustering is why Russian is typically described as having three of four classes: there are few general inflectional patterns, but many words with small deviations from these.

Clustering demonstrates one reason why node connectivity patterns affect system complexity. On the one hand, classes with high-density ties interfere with each other analogically. It might therefore seem that a greater density of edges in a network would lead monotonically to greater system complexity. However, when classes cluster, the interfering classes have mostly the same exponence. Strong clustering can thus actually lead to good interpredictability of forms for the majority of cells, even in a strongly connected network. It turns out there is no uniform relationship between the number of edges in a graph (or their weight) and the complexity of an inflectional system (Parker and Sims, to appear). This makes clustering an important network property for cross-linguistic comparison.

In an undirected network, the local clustering coefficient C_i of a node v_i with k neighbors is defined as:

$$C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

where N_i is the neighborhood of v_i, specifically, the set of nodes to which v_i is directly connected by an edge. The local clustering coefficient of v_i is thus the total number of edges among v_i's neighbors, divided by the total possible number of edges among neighbors. The global clustering coefficient of a system is the mean calculated over all C_i; values range between 0 and 1. The Russian nominal network has a global clustering coefficient of 0.816 (s.d. = 0.147).

5.4 Mean shortest path length

The path length between two nodes is the number of edges that must be followed to get from one to the other. Path length, like clustering coefficient, thus reflects patterns of network connectivity. Since edges in the inflectional network represent paths of analogical reasoning, the length of a path between a pair of nodes can be interpreted as being related to the likelihood of analogical interference between those classes, with low numbers indicating greater potential interference.

Since the Russian nominal network is not fully connected, the mean shortest path length for Russian nouns is here calculated within component. (Across components there are no paths, so shortest path length is infinite.) When calculated without edge weight (using a breadth-first search algorithm), the Russian network has a mean shortest path length of 1.249 (s.d. = 0.134) and when calculated taking edge weight into account (using the Dijkstra algorithm), the mean shortest path length is 8.929 (s.d. = 1.42).[10]

5.5 Betweenness centrality

We might also want to know which nodes are most central in the network. Central nodes are ones that are most likely to have shortest paths traverse them, often by virtue of them being connected to maximally separate parts of the network. As such, they are classes that are disproportionately likely to create pivots among classes that are more distinct, relative to other nodes in the net-

[8] The regression line excludes two nodes with degree of 1 and edge weight of 10. These are the same two nodes that belong to a separate component. If these are instead analyzed as a single class with a cross-cutting paradigm condition (Baerman et al., 2017), the merged class has degree of 0.

[9] Although there is not space in this paper to dive further into this issue, other languages show different degree-to-weight distributions.

[10] Shortest path length calculated over weighted edges seeks to minimize edge weight, treating edge weight as distance or cost. In the Russian nominal network, however, edge weight reflects similarity: more similar classes are connected by heavier edges. This would, oddly, result in the algorithm finding paths through maximally dissimilar classes. Edge weights were thus reversed for calculations of path length. Since Russian nouns have 12 cells, the maximum possible edge weight is 11. An edge weight of 11 was transformed to a value of 1, 10 was transformed to 2, etc.

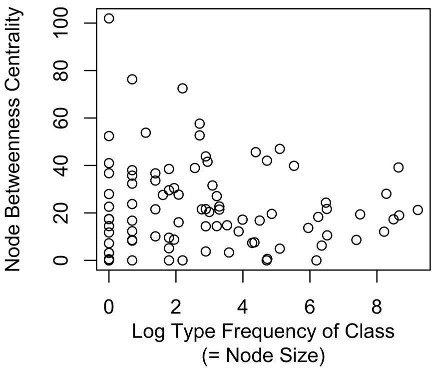


Figure 4: Correlation between node size and betweenness centrality for Russian nouns

work, putting those classes' exponents into potential analogical competition.

Betweenness centrality is calculated based on the set of shortest paths between v_i and v_j, for all possible values of i and j (where $i \neq j$). The betweenness centrality of a node v_k is the number of shortest paths in that set that include v_k, where $k \neq i, j$. In Figure 2 nodes are colored according to their betweenness centrality value, with darker red indicating more centrality. Figure 4 shows the betweenness centrality of classes as a function of their log type frequency.

Notice that low type frequency noun classes in Russian may be either high or low in centrality, but high type frequency classes have only low centrality. The nodes with the highest betweenness centrality turn out to be ones that are *mostly* regular but have irregularities that cross-cut the conventional classes in one or a few cells in the paradigm (especially, stress shift, vowel-zero alternation,[11] or an irregular nominative plural). Classes with the lowest betweenness centrality may also have low type frequency and exhibit irregularity, but in a different way: they are either uninflected or have unique stem extensions that serve to differentiate them from most other classes in most cells. Betweenness centrality thus reveals two different kinds of irregularity in Russian nouns, with different connectivity profiles within the network.

The distribution in Figure 4 is consistent with the observation by Sims and Parker (2016) that low type frequency classes contribute disproportionately to the unpredictability (complexity) of

[11]E.g. NOM.SG *otec* 'father', GEN.SG *otc-a*.

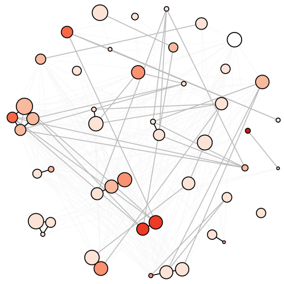


Figure 5: Inflection class system of Greek nouns

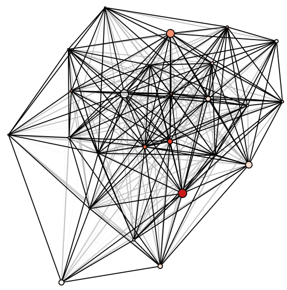


Figure 6: Inflection class system of Nuer nouns

the Russian nominal system; Stump and Finkel (2013) make a similar generalization based primarily on Icelandic verbs. However, it is seems likely that the true underlying issue has to do with how classes are embedded in their network – the effect is driven by classes with high betweenness centrality, which are themselves likely to have low type frequency.

6 Cross-linguistic comparison

I now turn to look at how these network measures might be used as a basis for typological comparison. Table 2 gives summary information

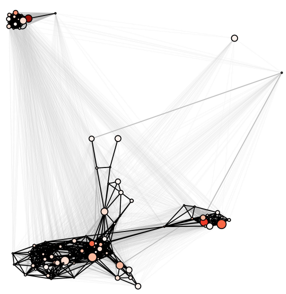


Figure 7: Inflection class system of Palantla Chinantec verbs

Language	Family	Cells	Classes	Lexemes	Sources
Chinantec verbs	Oto-Manguean	24	101	838	(Merrifield and Anderson, 2007)
French verbs	Indo-European	49	65	6,485	(Stump and Finkel, 2013)
Greek nouns	Indo-European	6	48	25,370	(Sims, 2015; Idryma Manoli Triantafyllidi, 1998)
Icelandic verbs	Indo-European	30	146	1,034	(Stump and Finkel, 2013; Jörg, 1989)
Kadiwéu verbs	Mataco-Guaicura	5	57	364	(Baerman et al., 2015; Griffiths, 2002)
Nuer nouns	Nilotic	6	25	252	(Baerman, 2012)
Russian nouns	Indo-European	12	87	43,486	(Parker, 2016; Zaliznjak, 1977)
Seri verbs	Isolate	4	254	952	(Baerman, 2016; Moser and Marlett, 2010)
Võro verbs	Uralic	9	23	4,668	(Baerman, 2014; Iva, 2007)

Table 2: Summary properties of the languages under investigation. Where more than one data sources is listed, the first is the direct source; the second is the original source

and sources for nine inflectional systems investigated here: Palantla Chinantec verbs, French verbs, Greek nouns, Icelandic verbs, Kadiwéu verbs, Nuer nouns, Russian nouns, Seri nouns, and Võro verbs. See Sims and Parker (2016) for further information about these data sets. This represents an opportunistic sample; it is not genetically or geographically balanced. This section focuses on comparing mean shortest path length and global clustering coefficient across these languages. A comparison based on the other metrics is left to future work for reasons of space, but the example is illustrative of how graph-theoretic measures can lead to new generalizations about the typological space of morphological systems.

Impressionistically, the diversity of the nine languages is striking. In addition to differing substantially in how many paradigm cells and classes they have, Figures 5 through 7 show the inflectional networks for Greek, Nuer, and Palantla Chinantec. The Greek nouns are connected by relatively fewer and weaker edges whereas the Nuer nouns are robustly connected. Additionally, nodes clusters into distinct groups in Palantla Chinantec, like in Russian.

Interestingly, however, when we turn to measures of shortest path length and clustering coefficient, an emergent pattern is evident. For shortest path length and clustering coefficient, direct comparison across languages is not meaningful because the sizes of the inflectional systems (number of nodes and edges) differ. More meaningful is a comparison between the inflectional systems and randomized versions of those systems. Simulated languages were generated by randomly sampling with replacement from the set of exponents for each paradigm cell, assigning them to classes. The exponents for each paradigm cell were sampled separately. The resulting simulated systems have the same number of allomorphs and classes as the real systems, but the paradigmatic relations that define the internal organization of the system have been randomly shuffled.

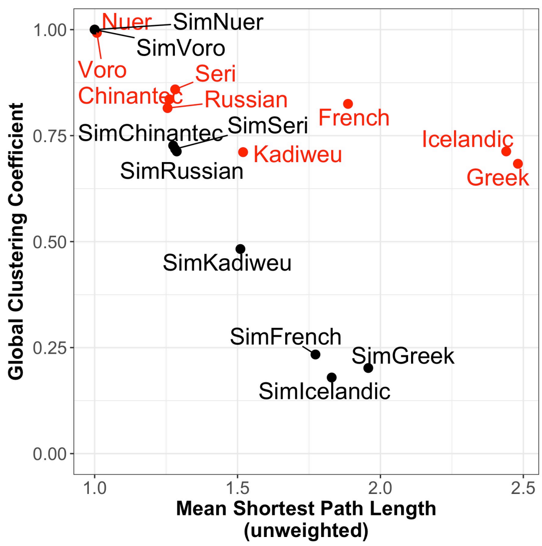


Figure 8: Comparison of real and simulated (resampled) inflection class systems according to mean shortest path length and global clustering coefficient

The results are shown in Figure 8.[12] (For the simulated languages, mean values from 100 ran-

[12] A version based on weighted edges, in which the distribution of weights from each real language was sampled with replacement and assigned at random to edges, produced qualitatively similar results.

domizations are shown.) The real systems differ from the simulated systems primarily in clustering, with the real languages exhibiting relatively more clustering as path length increases. Notably, for Nuer and Võro there is no meaningful difference between the real and simulated versions in either clustering or path length. This is equivalent to saying that Nuer and Võro lack (non-random) inflection class structure.

The closer the mean shortest path length of a network is to a value of 1, the closer that network necessarily is to forming a single large cluster, since every node is directly connected to every other node. This is what we see in Nuer and Võro. In contrast, networks with relatively long average path length values are relatively sparsely populated with edges (compare Figure 5 to Figure 6). In inflectional terms, this translates to classes that are more distinct. This sparsity gives more opportunity for (non-random) clustering. At the same time, it is not true that these networks *must* cluster to a significant degree, as the divergence between the real and the simulated languages shows.

The fact that in many languages, microclasses can be grouped into successively larger macroclasses is not a new observation (Brown and Hippisley, 2012; Dressler et al., 2006), but the generalization that some types of languages (i.e. ones whose networks are relatively sparsely populated with edges) are more likely to have this property is a new typological observation. But *why* do languages with greater average path length also employ significant amounts of clustering? Here it is not possible to do more than speculate in a broad way, but one possibility is that inflection classes that are more distinct are more likely to fracture over time as a result of independent changes (e.g. sound change), leaving groups of closely related but not identical classes. When classes are more distinct to begin with, such changes may be more likely to result in clustering. Further work would be needed to examine this possibility. But whatever the reason for the emergent pattern in Figure 8, it shows the ability of graph-theoretic measures, when applied to inflectional typology, to unearth new empirical generalizations about the internal organization of inflectional systems.

7 Conclusions

While traditional approaches to inflectional typology have focused on the size of inflectional systems, this does not capture their internal organization, particularly as related to the predictability of inflected forms (also called the system's I-complexity). I have argued for thinking of inflectional systems as networks in which the nodes are classes and the edges are exponents that two classes have in common. This allows for tools from graph theory to be applied to the task of describing the internal organization of inflectional systems in their full richness.

The cross-linguistic comparison in section 6 highlighted the possibility of using graph-theoretic measures to compare the network structure of inflection class systems. The measures employed here offer a fundamentally different basis for typology than in traditional approaches and revealed novel generalizations about the typological space of morphological systems. In particular, clustering emerged as a common property.

Future work should focus on identifying which graph-theoretic measures are most useful for cross-linguistic comparison of morphological systems. Additionally, as has already been demonstrated in other domains (e.g. transportation networks), node connectivity profiles not only define classes of networks, but affect the dynamics of a network differently (Guimerà et al., 2007). This hints at the possibility of better predicting inflectional change. Ultimately, graph theory offers a promising basis for inflectional typology, and more.

Acknowledgments

Preliminary versions of this work were presented at the Future of Language Science event (Northwestern University) and Linguistic Institute Sunday Poster Sessions (University of Kentucky), and to the Laboratoire de Linguistique Formelle (Université de Paris and CNRS). I thank the audiences in those places for their helpful feedback. Conversations with Jeff Parker have shaped my thinking and I thank him for his work compiling some of the data sets. I also thank Matthew Baerman, Raphael Finkel and Greg Stump for making data sets available. This paper has benefited from the comments from three anonymous reviewers; all errors remain my own.

References

Farrell Ackerman, James P. Blevins, and Robert Malouf. 2009. Parts and wholes: Implicative patterns in inflectional paradigms. In James P. Blevins and Juliette Blevins, editors, *Analogy in grammar: Form and acquisition*, pages 54–82. Oxford University Press.

Farrell Ackerman and Robert Malouf. 2013. Morphological organization: The low conditional entropy conjecture. *Language*, 89(3):429–464.

Farrell Ackerman, Robert Malouf, and James P. Blevins. 2016. Patterns and discriminability in language analysis. *Word Structure*, 9(2):132–155.

Matthew Baerman. 2012. Paradigmatic chaos in Nuer. *Language*, 88(3):467–494.

Matthew Baerman. 2014. Covert systematicity in a distributionally complex system. *Journal of Linguistics*, 50(1):1–47.

Matthew Baerman. 2016. Seri verb classes: Morphosyntactic motivation and morphological autonomy. *Language*, 92(4):792–823.

Matthew Baerman, Dunstan Brown, and Greville G. Corbett. 2017. *Morphological complexity*. Cambridge University Press.

Matthew Baerman, Dunstan Brown, Roger Evans, Greville G. Corbett, and Lynne Cahill. 2015. Surrey morphological complexity database.

Sacha Beniamine. 2018. *Classifications flexionnelles. Étude quatitative des structures de paradigmes*. Ph.D. thesis, Linguistique, Université Sorbonne Paris Cité – Université Paris Diderot (Paris 7).

Sacha Beniamine, Olivier Bonami, and Joyce McDonough. 2017a. When segmentation helps: Implicative structure and morph boundaries in the Navajo verb. In *Abstracts from the First International Symposium of Morphology, University of Lille*, pages 11–15.

Sacha Beniamine, Olivier Bonami, and Benoît Sagôt. 2017b. Inferring inflection classes with description length. *Journal of Language Modelling*, 5(3):465–525.

Ginestra Bianchoni. 2018. *Multilayer networks: Structure and function*. Oxford University Press.

James P. Blevins. 2006. Word-based morphology. *Journal of Linguistics*, 42:531–573.

James P. Blevins. 2016. *Word and paradigm morphology*. Oxford University Press.

Olivier Bonami and S. Beniamine. 2016. Joint predictiveness in inflectional paradigms. *Word Structure*, 9:156–182.

Dunstan Brown and Andrew Hippisley. 2012. *Network Morphology: A defaults-based theory of word structure*. Cambridge University Press.

Bernard Comrie. 1981. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago Press.

Greville G. Corbett. 1982. Gender in Russian: An account of gender specification and its relationship to declension. *Russian Linguistics*, 6:197–232.

Ryan Cotterell, Christo Kirov, Mans Hulden, and Jason Eisner. to appear. On the complexity and typology of inflectional morphological systems. *Transactions of the Association for Computational Linguistics*.

Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

Wolfgang U. Dressler, Marianne Kilani-Schoch, Natalia Gagarina, Lina Pestal, and Markus Pöchtrager. 2006. On the typology of inflection class systems. *Folia Linguistica*, 40:51–74.

Micha Elsner, Andrea D. Sims, Alex Erdmann, Antonio Hernandez, Evan Jaffe, Lifeng Jin, Martha Johnson, Shuan Karim, David King, Luana Lamberti Nunes, Nathan Rasmussen, Cory Shain, Symon Stevens Guille, Stephanie Antetomaso, Kendra Dickinson, Noah Diewald, Michelle McKenzie, and Byung-Doh Oh. submitted. Models of morphological learning: Implications for typology and change.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proc. of NAACL*.

John A. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

John A. Goldsmith. 2010. Segmentation and morphology. In Alexander Clark, Chris Fox, and Shalom Lappin, editors, *The handbook of computational linguistics and natural language processing*, pages 364–393. Wiley-Blackwell.

Glyn Griffiths. 2002. *Dicionário da lingua Kadiwéu*. Summer Institute of Linguistics.

Roger Guimerà, Martha Sales-Pardo, and Luís A.N. Amaral. 2007. Classes of complex networks defined by role-to-role connectivity profiles. *Nature Physics*, 3:63–69.

Zellig S. Harris. 1970. From phoneme to morpheme. In Zellig S. Harris, editor, *Papers in structural and transformational linguistics*, pages 32–67. Springer.

Charles F. Hockett. 1947. Problems of morphemic analysis. *Language*, 23:321–343.

Idryma Manoli Triantafyllidi, editor. 1998. *Lexiko tis koinis neoellinikis*. Aristoteleio Panepistimio Thessalonikis, Institouto Neoellinikon Epoudon.

Sulev Iva. 2007. *Võru kirjakeele sõnamuutmissüsteem*. University of Tartu.

Christine Jörg. 1989. *Isländische Konjugationstabellen / Icelandic conjugation tables / Tableaux de conjugaison islandaise / Beygingatöflur islenskra sagna*. H. Buske.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. *ACL 2016*, page 62.

Constantine Lignos and Charles Yang. 2016. Morphology and language acquisition. In Andrew Hippisley and Gregory T. Stump, editors, *The Cambridge handbook of morphology*, page 743764. Cambridge University Press.

Robert Malouf. 2017. Abstractive morphological learning with a recurrent neural network. *Morphology*, 27(4):431–458.

Christopher D. Manning. 1998. The segmentation problem in morphology learning. In D.M.W. Powers, editor, *Proceedings of NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Language Learning*, pages 299–305. Association for Computational Linguistics.

John Mansfield. 2016. Intersecting formatives and inflectional predictability: How do speakers and learners predict the correct form of Murrinhpatha verbs? *Word Structure*, 9:183–214.

P.H. Matthews. 1972. *Inflectional morphology: A theoretical study based on aspects of Latin verb conjugation*. Cambridge University Press.

William R. Merrifield and Alfredo B. Anderson. 2007. *Diccionario Chinanteco de la diáspora del pueblo antiguo de San Pedro Tlatepuzco, Oaxaca*. Instituto Lingüístico de Verano.

Mary B. Moser and Stephen A. Marlett. 2010. *Comcaac quih yaza quih hand ihiip hav = Diccionario Seri-Español-Inglés*. Plaza y Valdés Editores y Universidad de Sonora.

Eugene A. Nida. 1949. *Morphology: The descriptive analysis of words*. University of Michigan Press.

Jeff Parker. 2016. *Inflectional complexity and cognitive processing: An experimental and corpus-based investigation of Russian*. Ph.D. thesis, Department of Slavic and East European Languages and Cultures, The Ohio State University.

Jeff Parker and Andrea D. Sims. to appear. Irregularity, paradigmatic layers, and the complexity of inflection class systems: A study of Russian nouns. In Peter Arkadiev and Francesco Gardani, editors, *Complexities of morphology*. Oxford University Press.

Frans Plank. 1999. Split morphology: How agglutination and flexion mix. *Linguistic Typology*, 3:279–340.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

Miikka Silfverberg and Mans Hulden. 2018. An encoder-decoder approach to the paradigm cell filling problem. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2889.

Andrea D. Sims. 2015. *Inflectional defectiveness*. Cambridge University Press.

Andrea D. Sims and Jeff Parker. 2016. How inflection class systems work: On the informativity of implicative structure. *Word Structure*, 9:215–239.

Andrew Spencer. 2012. Identifying stems. *Word Structure*, 5:88–108.

Gregory T. Stump and Raphael Finkel. 2013. *Morphological typology: From word to paradigm*. Cambridge University Press.

V.V. Vinogradov, E.S. Istrina, and S.G. Barxudarov. 1952. *Grammatika russkogo jazyka 1: Fonetika i morfologija*. Akademija Nauk SSSR.

Wolfgang U. Wurzel. 1989. *Inflectional morphology and naturalness*. Kluwer.

A.A. Zaliznjak. 1977. *Grammatičeskij slovar' russkogo jazyka: Slovoizmenenie*. Russkij jazyk.

Acquisition of Inflectional Morphology in Artificial Neural Networks With Prior Knowledge

Katharina Kann
New York University, USA
kann@nyu.edu

Abstract

How does knowledge of one language's morphology influence learning of inflection rules in a second one? In order to investigate this question in artificial neural network models, we perform experiments with a sequence-to-sequence architecture, which we train on different combinations of eight source and three target languages. A detailed analysis of the model outputs suggests the following conclusions: (i) if source and target language are closely related, acquisition of the target language's inflectional morphology constitutes an easier task for the model; (ii) knowledge of a prefixing (resp. suffixing) language makes acquisition of a suffixing (resp. prefixing) language's morphology more challenging; and (iii) surprisingly, a source language which exhibits an agglutinative morphology simplifies learning of a second language's inflectional morphology, independent of their relatedness.

1 Introduction

A widely agreed-on fact in language acquisition research is that learning of a second language (L2) is influenced by a learner's native language (L1) (Dulay and Burt, 1974; Kellerman, 1979). A language's morphosyntax seems to be no exception to this rule (Bliss, 2006), but the exact nature of this influence remains unknown. For instance, it is unclear whether it is constraints imposed by the phonological or by the morphosyntactic attributes of the L1 that are more important during the process of learning an L2's morphosyntax.

Within the area of natural language processing (NLP) research, experimenting on neural network models just as if they were human subjects has recently been gaining popularity (Ettinger et al., 2016, 2017; Kim et al., 2019). Often, so-called probing tasks are used, which require a specific subset of linguistic knowledge and can, thus, be leveraged for qualitative evaluation. The goal is to answer the question: What do neural networks learn that helps them to succeed in a given task?

	walk	**eat**
`Inf`	dance	eat
`3rdSgPres`	dances	eats
`PresPart`	dancing	eating
`Past`	danced	ate
`PastPart`	danced	eaten

Table 1: Paradigms of the English lemmas *dance* and *eat*. *dance* has 4 distinct inflected forms; *eat* has 5.

Neural network models, and specifically sequence-to-sequence models, have pushed the state of the art for morphological inflection – the task of learning a mapping from lemmata to their inflected forms – in the last years (Cotterell et al., 2016). Thus, in this work, we experiment on such models, asking not *what* they learn, but, motivated by the respective research on human subjects, the related question of *how what they learn depends on their prior knowledge*. We manually investigate the errors made by artificial neural networks for morphological inflection in a target language after pretraining on different source languages. We aim at finding answers to two main questions: (i) Do errors systematically differ between source languages? (ii) Do these differences seem explainable, given the properties of the source and target languages? In other words, we are interested in exploring if and how L2 acquisition of morphological inflection depends on the L1, i.e., the "native language", in neural network models.

To this goal, we select a diverse set of eight source languages from different language families – Basque, French, German, Hungarian, Italian, Navajo, Turkish, and Quechua – and three target languages – English, Spanish and Zulu. We pretrain a neural sequence-to-sequence architecture on each of the source languages and then fine-tune the resulting models on small datasets in each of the target languages. Analyzing the errors made by the systems, we find that (i) source and target language being closely related simplifies the successful learning of inflection in the target language, (ii) the task is harder to learn in a prefixing language if the source language is suffixing – as well as the other way around, and (iii) a source language which exhibits an agglutinative morphology simplifies learning of a second language's inflectional morphology.

2 Task

Many of the world's languages exhibit rich inflectional morphology: the surface form of an individual lexical entry changes in order to express properties such as person, grammatical gender, or case. The citation form of a lexical entry is referred to as the *lemma*. The set of all possible surface forms or *inflections* of a lemma is called its *paradigm*. Each inflection within a paradigm can be associated with a tag, i.e., `3rdSgPres` is the morphological tag associated with the inflection *dances* of the English lemma *dance*. We display the paradigms of *dance* and *eat* in Table 1.

The presence of rich inflectional morphology is problematic for NLP systems as it increases word form sparsity. For instance, while English verbs can have up to 5 inflected forms, Archi verbs have thousands (Kibrik, 1998), even by a conservative count. Thus, an important task in the area of morphology is morphological inflection (Durrett and DeNero, 2013; Cotterell et al., 2018), which consists of mapping a lemma to an indicated inflected form. An (irregular) English example would be

$$(\text{eat}, \texttt{PAST}) \rightarrow \text{ate}$$

with `PAST` being the target tag, denoting the past tense form. Additionally, a rich inflectional morphology is also challenging for L2 language learners, since both rules and their exceptions need to be memorized.

In NLP, morphological inflection has recently frequently been cast as a sequence-to-sequence problem, where the sequence of target (sub-)tags together with the sequence of input characters constitute the input sequence, and the characters of the inflected word form the output. Neural models define the state of the art for the task and obtain high accuracy if an abundance of training data is available. Here, we focus on learning of inflection from limited data if information about another language's morphology is already known. We, thus, loosely simulate an L2 learning setting.

Formal definition. Let $\mathcal{M}$ be the paradigm slots which are being expressed in a language, and w a lemma in that language. We then define the paradigm π of w as:

$$\pi(w) = \Big\{\big(f_k[w], t_k\big)\Big\}_{k \in \mathcal{M}(w)} \qquad (1)$$

$f_k[w]$ denotes an inflected form corresponding to tag t_k, and w and $f_k[w]$ are strings consisting of letters from an alphabet Σ.

The task of morphological inflection consists of predicting a missing form $f_i[w]$ from a paradigm, given the lemma w together with the tag t_i.

3 Model

3.1 Pointer–Generator Network

The models we experiment with are based on a pointer–generator network architecture (Gu et al., 2016; See et al., 2017), i.e., a recurrent neural network (RNN)-based sequence-to-sequence network with attention and a copy mechanism. A standard sequence-to-sequence model (Bahdanau et al., 2015) has been shown to perform well for morphological inflection (Kann and Schütze, 2016) and has, thus, been subject to cognitively motivated experiments (Kirov and Cotterell, 2018) before. Here, however, we choose the pointer–generator variant of Sharma et al. (2018), since it performs better in low-resource settings, which we will assume for our target languages. We will explain the model shortly in the following and refer the reader to the original paper for more details.

Encoders. Our architecture employs two separate encoders, which are both bi-directional long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997): The first processes the morphological tags which describe the desired target form one by one.[1] The second encodes the sequence of characters of the input word.

[1]In contrast to other work on cross-lingual transfer in deep learning models we do not employ language embeddings.

Attention. Two separate attention mechanisms are used: one per encoder LSTM. Taking all respective encoder hidden states as well as the current decoder hidden state as input, each of them outputs a so-called context vector, which is a weighted sum of all encoder hidden states. The concatenation of the two individual context vectors results in the final context vector c_t, which is the input to the decoder at time step t.

Decoder. Our decoder consists of a unidirectional LSTM. Unlike a standard sequence-to-sequence model, a pointer–generator network is not limited to generating characters from the vocabulary to produce the output. Instead, the model gives certain probability to copying elements from the input over to the output. The probability of a character y_t at time step t is computed as a sum of the probability of y_t given by the decoder and the probability of copying y_t, weighted by the probabilities of generating and copying:

$$p(y_t) = \alpha p_{\text{dec}}(y_t) + (1 - \alpha) p_{\text{copy}}(y_t) \quad (2)$$

$p_{\text{dec}}(y_t)$ is calculated as an LSTM update and a projection of the decoder state to the vocabulary, followed by a softmax function. $p_{\text{copy}}(y_t)$ corresponds to the attention weights for each input character. The model computes the probability α with which it generates a new output character as

$$\alpha = \sigma(w_c c_t + w_s s_t + w_y y_{t-1} + b) \quad (3)$$

for context vector c_t, decoder state s_t, embedding of the last output y_{t-1}, weights w_c, w_s, w_y, and bias vector b. It has been shown empirically that the copy mechanism of the pointer–generator network architecture is beneficial for morphological generation in the low-resource setting (Sharma et al., 2018).

3.2 Pretraining and Finetuning

Pretraining and successive fine-tuning of neural network models is a common approach for handling of low-resource settings in NLP. The idea is that certain properties of language can be learned either from raw text, related tasks, or related languages. Technically, *pretraining* consists of estimating some or all model parameters on examples which do not necessarily belong to the final target task. *Fine-tuning* refers to continuing training of such a model on a target task, whose data is often limited. While the sizes of the pretrained model parameters usually remain the same between the two phases, the learning rate or other details of the training regime, e.g., dropout, might differ. Pretraining can be seen as finding a suitable initialization of model parameters, before training on limited amounts of task- or language-specific examples.

In the context of morphological generation, pretraining in combination with fine-tuning has been used by Kann and Schütze (2018), which proposes to pretrain a model on general inflection data and fine-tune on examples from a specific paradigm whose remaining forms should be automatically generated. Famous examples for pretraining in the wider area of NLP include BERT (Devlin et al., 2019) or GPT-2 (Radford et al., 2019): there, general properties of language are learned using large unlabeled corpora.

Here, we are interested in pretraining as a simulation of familiarity with a native language. By investigating a fine-tuned model we ask the question: How does extensive knowledge of one language influence the acquisition of another?

4 Experimental Design

4.1 Target Languages

We choose three target languages.

English (ENG) is a morphologically impoverished language, as far as inflectional morphology is concerned. Its verbal paradigm only consists of up to 5 different forms and its nominal paradigm of only up to 2. However, it is one of the most frequently spoken and taught languages in the world, making its acquisition a crucial research topic.

Spanish (SPA), in contrast, is morphologically rich, and disposes of much larger verbal paradigms than English. Like English, it is a suffixing language, and it additionally makes use of internal stem changes (e.g., *o* $\rightarrow$ *ue*).

Since English and Spanish are both Indo-European languages, and, thus, relatively similar, we further add a third, unrelated target language. We choose Zulu (ZUL), a Bantoid language. In contrast to the first two, it is strongly prefixing.

4.2 Source Languages

For pretraining, we choose languages with different degrees of relatedness and varying morphological similarity to English, Spanish, and Zulu. We

	ENG	SPA	ZUL	EUS	FRA	DEU	HUN	ITA	NAV	TUR	QVH
20A Fusion of Selected Inflectional Formatives	0	0	0	0	0	0	0	1	0	0	0
21A Exponence of Selected Inflectional Formatives	0	1	0	1	0	2	1	3	3	1	1
21B Exponence of Tense-Aspect-Mood Inflection	0	1	0	0	1	0	0	2	2	0	0
22A Inflectional Synthesis of the Verb	0	1	1	1	1	0	1	2	2	3	4
23A Locus of Marking in the Clause	0	1	2	1	3	0	0	4	4	0	0
24A Locus of Marking in Possessive Noun Phrases	0	0	0	0	0	0	0	1	1	2	0
25A Locus of Marking: Whole-language Typology	0	1	1	1	1	0	1	2	2	1	0
25B Zero Marking of A and P Arguments	0	0	0	0	0	0	0	1	1	0	0
26A Prefixing vs. Suffixing in Inflectional Morphology	0	0	1	2	0	0	0	0	1	0	0
27A Reduplication	0	0	1	2	0	0	2	0	0	2	1
28A Case Syncretism	0	1	2	0	1	1	3	4	2	3	3
29A Syncretism in Verbal Person/Number Marking	0	0	0	1	0	0	1	2	1	1	1

Table 2: WALS features from the *Morphology* category. 20A: 0=Exclusively concatenative, 1=N/A. 21A: 0=No case, 1=Monoexponential case, 2=Case+number, 3=N/A. 21B: 0=monoexponential TAM, 1=TAM+agreement, 2=N/A. 22A: 0=2-3 categories per word, 1=4-5 categories per word, 2=N/A, 3=6-7 categories per word, 4=8-9 categories per word. 23A: 0=Dependent marking, 1=Double marking, 2=Head marking, 3=No marking, 4=N/A. 24A: 0=Dependent marking, 1=N/A, 2=Double marking. 25A: 0=Dependent-marking, 1=Inconsistent or other, 2=N/A. 25B: 0=Non-zero marking, 1=N/A. 26A: 0=Strongly suffixing, 1=Strong prefixing, 2=Equal prefixing and suffixing. 27A: 0=No productive reduplication, 1=Full reduplication only, 2=Productive full and partial reduplication. 28A: 0=Core cases only, 1=Core and non-core, 2=No case marking, 3=No syncretism, 4=N/A. 29A: 0=Syncretic, 1=Not syncretic, 2=N/A.

limit our experiments to languages which are written in Latin script.

As an estimate for morphological similarity we look at the features from the *Morphology* category mentioned in The World Atlas of Language Structures (WALS).[2] An overview of the available features as well as the respective values for our set of languages is shown in Table 2.

We decide on Basque (EUS), French (FRA), German (DEU), Hungarian (HUN), Italian (ITA), Navajo (NAV), Turkish (TUR), and Quechua (QVH) as source languages.

Basque is a language isolate. Its inflectional morphology makes similarly frequent use of prefixes and suffixes, with suffixes mostly being attached to nouns, while prefixes and suffixes can both be employed for verbal inflection.

French and Italian are Romance languages, and thus belong to the same family as the target language Spanish. Both are suffixing and fusional languages.

German, like English, belongs to the Germanic language family. It is a fusional, predominantly suffixing language and, similarly to Spanish, makes use of stem changes.

Hungarian, a Finno-Ugric language, and Turkish, a Turkic language, both exhibit an agglutinative morphology, and are predominantly suffixing. They further have vowel harmony systems.

Navajo is an Athabaskan language and the only source language which is strongly prefixing. It further exhibits consonant harmony among its sibilants (Rice, 2000; Hansson, 2010).

Finally, Quechua, a Quechuan language spoken in South America, is again predominantly suffixing and unrelated to all of our target languages.

4.3 Hyperparameters and Data

We mostly use the default hyperparameters by Sharma et al. (2018).[3] In particular, all RNNs have one hidden layer of size 100, and all input and output embeddings are 300-dimensional.

For optimization, we use ADAM (Kingma and Ba, 2014). Pretraining on the source language is done for exactly 50 epochs. To obtain our final models, we then fine-tune different copies of each pretrained model for 300 additional epochs for each target language. We employ dropout (Srivastava et al., 2014) with a coefficient of 0.3 for pretraining and, since that dataset is smaller, with a coefficient of 0.5 for fine-tuning.

We make use of the datasets from the CoNLL–SIGMORPHON 2018 shared task (Cotterell et al., 2018). The organizers provided a low, medium, and high setting for each language, with 100, 1000, and 10000 examples, respectively. For all L1 languages, we train our models on the high-resource datasets with 10000 examples. For fine-

[2] `https://wals.info`

[3] `github.com/abhishek0318/conll-sigmorphon-2018`

	EUS	FRA	DEU	HUN	ITA	NAV	TUR	QVH
ENG	45.8	76.1	82.0	85.6	84.7	53.2	81.7	68.3
SPA	23.9	53.3	53.8	58.2	56.9	33.1	52.0	49.0
ZUL	10.8	17.1	23.0	23.0	21.9	13.6	24.9	10.7

Table 3: Test accuracy.

	EUS	FRA	DEU	HUN	ITA	NAV	TUR	QVH
ENG	44.2	75.8	81.4	84.5	84.3	50.8	81.6	67.3
SPA	24.5	55.1	54.8	61.0	58.3	33.6	51.9	51.8
ZUL	12.4	21.8	24.5	25.7	22.2	13.8	28.7	12.2

Table 4: Validation accuracy.

tuning, we use the low-resource datasets.

5 Quantitative Results

In Table 3, we show the final test accuracy for all models and languages. Pretraining on EUS and NAV results in the weakest target language inflection models for ENG, which might be explained by those two languages being unrelated to ENG and making at least partial use of prefixing, while ENG is a suffixing language (cf. Table 2). In contrast, HUN and ITA yield the best final models for ENG. This is surprising, since DEU is the language in our experiments which is closest related to ENG.

For SPA, again HUN performs best, followed closely by ITA. While the good performance of HUN as a source language is still unexpected, ITA is closely related to SPA, which could explain the high accuracy of the final model. As for ENG, pretraining on EUS and NAV yields the worst final models – importantly, accuracy is over 15% lower than for QVH, which is also an unrelated language. This again suggests that the prefixing morphology of EUS and NAV might play a role.

Lastly, for ZUL, all models perform rather poorly, with a minimum accuracy of 10.7 and 10.8 for the source languages QVH and EUS, respectively, and a maximum accuracy of 24.9 for a model pretrained on Turkish. The latter result hints at the fact that a regular and agglutinative morphology might be beneficial in a source language – something which could also account for the performance of models pretrained on HUN.

6 Qualitative Results

For our qualitative analysis, we make use of the validation set. Therefore, we show validation set accuracies in Table 4 for comparison. As we can see, the results are similar to the test set results for all language combinations. We manually annotate the outputs for the first 75 development examples for each source–target language combination. All found errors are categorized as belonging to one of the following categories.

Stem Errors

- **SUB(X)**: This error consists of a wrong substitution of one character with another. SUB(V) and SUB(C) denote this happening with a vowel or a consonant, respectively. Letters that differ from each other by an accent count as different vowels.
  Example: *decultared* instead of *decultured*

- **DEL(X)**: This happens when the system ommits a letter from the output. DEL(V) and DEL(C) refer to a missing vowel or consonant, respectively.
  Example: *firte* instead of *firtle*

- **NO_CHG(X)**: This error occurs when inflecting the lemma to the gold form requires a change of either a vowel (NO_CHG(V)) or a consonant (NO_CHG(C)), but this is missing in the predicted form.
  Example: *verto* instead of *vierto*

- **MULT**: This describes cases where two or more errors occur in the stem. Errors concerning the affix are counted for separately.
  Example: *aconcoonaste* instead of *acondicionaste*

- **ADD(X)**: This error occurs when a letter is mistakenly added to the inflected form. ADD(V) refers to an unnecessary vowel, ADD(C) refers to an unnecessary consonant.
  Example: *compillan* instead of *compilan*

- **CHG2E(X)**: This error occurs when inflecting the lemma to the gold form requires a change of either a vowel (CHG2E(V)) or a consonant (CHG2E(C)), and this is done, but the resulting vowel or consonant is incorrect.
  Example: *propace* instead of *propague*

Affix Errors

- **AFF**: This error refers to a wrong affix. This can be either a prefix or a suffix, depending on the correct target form.
  Example: *ezoJulayi* instead of *esikaJulayi*

	EUS	FRA	DEU	HUN	ITA	NAV	QVH	TUR
SUB(V)	2	2	0	2	2	2	0	3
DEL(C)	5	2	1	1	1	8	2	1
DEL(V)	6	1	2	0	2	5	4	1
NO_CHG(V)	1	1	0	1	1	2	3	1
MULT	18	3	3	0	1	13	13	0
ADD(V)	0	0	0	0	0	2	0	0
CHG2E(V)	0	0	0	0	0	0	0	0
ADD(C)	5	0	0	0	0	3	0	0
CHG2E(C)	0	0	0	0	0	0	0	0
NO_CHG(C)	0	0	0	0	0	0	0	0
AFF	10	8	3	5	5	9	9	8
CUT	0	0	1	0	0	0	0	0
REFL	0	0	0	0	0	0	0	0
REFL_LOC	0	0	0	0	0	0	0	0
OVERREG	1	1	1	1	1	1	1	1
Stem	37	9	6	4	7	35	22	6
Affix	10	8	4	5	5	9	9	8
Misc	1	1	1	1	1	1	1	1

Table 5: Error analysis for ENG as the model's L2.

- **CUT**: This consists of cutting too much of the lemma's prefix or suffix before attaching the inflected form's prefix or suffix, respectively.
  Example: *irradiseis* instead of *irradiaseis*

Miscellaneous Errors

- **REFL**: This happens when a reflective pronoun is missing in the generated form.
  Example: *doliéramos* instead of *nos doliéramos*

- **REFL_LOC**: This error occurs if the reflective pronouns appears at an unexpected position within the generated form.
  Example: *taparsebais* instead of *os tapabais*

- **OVERREG**: Overregularization errors occur when the model predicts a form which would be correct if the lemma's inflections were regular but they are not.
  Example: *underteach* instead of *undertaught*

6.1 Error Analysis: English

Table 5 displays the errors found in the 75 first ENG development examples, for each source language. From Table 4, we know that HUN > ITA > TUR > DEU > FRA > QVH > NAV > EUS, and we get a similar picture when analyzing the first examples. Thus, especially keeping HUN and TUR in mind, we cautiously propose a first conclusion: *familiarity with languages which exhibit an agglutinative morphology simplifies learning of a new language's morphology*.

Looking at the types of errors, we find that EUS and NAV make the most stem errors. For QVH we find less, but still over 10 more than for the remaining languages. This makes it seem that models pretrained on prefixing or partly prefixing languages indeed have a harder time to learn ENG inflectional morphology, and, in particular, to copy the stem correctly. Thus, our second hypotheses is that *familiarity with a prefixing language might lead to suspicion of needed changes to the part of the stem which should remain unaltered in a suffixing language*. DEL(X) and ADD(X) errors are particularly frequent for EUS and NAV, which further suggests this conclusion.

Next, the relatively large amount of stem errors for QVH leads to our second hypothesis: *language relatedness does play a role when trying to produce a correct stem of an inflected form*. This is also implied by the number of MULT errors for EUS, NAV and QVH, as compared to the other languages.

Considering errors related to the affixes which have to be generated, we find that DEU, HUN and ITA make the fewest. This further suggests the conclusion that, especially since DEU is the language which is closest related to ENG, *language relatedness plays a role for producing suffixes of inflected forms* as well.

Our last observation is that many errors are not found at all in our data sample, e.g., CHG2E(X) or NO_CHG(C). This can be explained by ENG having a relatively poor inflectional morphology, which does not leave much room for mistakes.

6.2 Error Analysis: Spanish

The errors committed for SPA are shown in Table 6, again listed by source language. Together with Table 4 it gets clear that SPA inflectional morphology is more complex than that of ENG: systems for all source languages perform worse.

Similarly to ENG, however, we find that most stem errors happen for the source languages EUS and NAV, which is further evidence for our previous hypothesis that *familiarity with prefixing languages impedes acquisition of a suffixing one*. Especially MULT errors are much more frequent for EUS and NAV than for all other languages. ADD(X) happens a lot for EUS, while ADD(C) is also frequent for NAV. Models pretrained on either

	EUS	FRA	DEU	HUN	ITA	NAV	QVH	TUR
SUB(V)	7	1	4	4	3	4	3	4
DEL(C)	4	0	0	0	0	1	1	0
DEL(V)	4	0	1	0	0	2	0	0
NO_CHG(V)	6	7	6	5	5	3	5	6
MULT	8	2	0	0	0	9	0	2
ADD(V)	4	2	0	0	0	0	1	0
CHG2E(V)	1	0	0	1	0	1	1	0
ADD(C)	3	1	1	0	0	3	0	1
CHG2E(C)	0	0	1	0	0	1	0	0
NO_CHG(C)	0	0	0	0	1	0	1	0
AFF	35	29	27	23	26	35	31	30
CUT	9	1	2	1	1	8	3	1
REFL	2	0	2	0	1	2	1	1
REFL_LOC	0	2	0	2	1	0	1	1
OVERREG	0	0	0	0	0	0	0	0
Stem	37	13	13	10	9	24	12	13
Affix	44	30	29	24	27	43	34	31
Misc	2	2	2	2	2	2	2	2

Table 6: Error analysis for SPA as the model's L2.

	EUS	FRA	DEU	HUN	ITA	NAV	QVH	TUR
SUB(V)	3	2	1	3	0	6	7	1
DEL(C)	4	6	1	4	6	3	2	2
DEL(V)	1	7	0	2	2	0	3	1
NO_CHG(V)	2	0	0	0	0	1	1	0
MULT	30	8	13	10	11	21	31	9
ADD(V)	0	1	1	3	1	2	0	2
CHG2E(V)	0	0	0	0	0	0	0	0
ADD(C)	1	3	1	6	4	2	1	1
CHG2E(C)	0	0	0	0	0	0	0	0
NO_CHG(C)	0	2	1	1	1	0	0	1
AFF	59	52	52	53	53	55	57	52
CUT	1	3	2	5	3	2	3	4
REFL	0	0	0	0	0	0	0	0
REFL_LOC	0	0	0	0	0	0	0	0
OVERREG	0	0	0	0	0	0	0	0
Stem	41	29	18	29	25	35	45	17
Affix	60	55	54	58	56	57	60	56
Misc	0	0	0	0	0	0	0	0

Table 7: Error analysis for ZUL as the model's L2.

language have difficulties with vowel changes, which reflects in NO_CHG(V). Thus, we conclude that this phenomenon is generally hard to learn.

Analyzing next the errors concerning affixes, we find that models pretrained on HUN, ITA, DEU, and FRA (in that order) commit the fewest errors. This supports two of our previous hypotheses: First, given that ITA and FRA are both from the same language family as SPA, *relatedness seems to be benficial for learning of the second language*. Second, the system pretrained on HUN performing well suggests again that *a source language with an agglutinative, as opposed to a fusional, morphology seems to be beneficial* as well.

6.3 Error Analysis: Zulu

In Table 7, the errors for Zulu are shown, and Table 4 reveals the relative performance for different source languages: TUR > HUN > DEU > ITA > FRA > NAV > EUS > QVH. Again, TUR and HUN obtain high accuracy, which is an additional indicator for our hypothesis that *a source language with an agglutinative morphology facilitates learning of inflection in another language*.

Besides that, results differ from those for ENG and SPA. First of all, more mistakes are made for all source languages. However, there are also several finer differences. For ZUL, the model pretrained on QVH makes the most stem errors, in particular 4 more than the EUS model, which comes second. Given that ZUL is a prefixing language and QVH is suffixing, this relative order seems important. QVH also committs the highest number of MULT errors.

The next big difference between the results for ZUL and those for ENG and SPA is that DEL(X) and ADD(X) errors, which previously have mostly been found for the prefixing or partially prefixing languages EUS and NAV, are now most present in the outputs of *suffixing* languages. Namely, DEL(C) occurs most for FRA and ITA, DEL(V) for FRA and QVH, and ADD(C) and ADD(V) for HUN. While some deletion and insertion errors are subsumed in MULT, this does not fully explain this difference. For instance, QVH has both the second most DEL(V) and the most MULT errors.

The overall number of errors related to the affix seems comparable between models with different source languages. This weakly supports the hypothesis that *relatedness reduces affix-related errors*, since none of the pretraining languages in our experiments is particularly close to ZUL. However, we do find more CUT errors for HUN and TUR: again, these are suffixing, while CUT for the target language SPA mostly happened for the prefixing languages EUS and NAV.

6.4 Limitations

A limitation of our work is that we only include languages that are written in Latin script. An interesting question for future work might, thus, regard the effect of disjoint L1 and L2 alphabets.

Furthermore, none of the languages included in our study exhibits a templatic morphology. We make this choice because data for templatic languages is currently mostly available in non-Latin alphabets. Future work could investigate languages with templatic morphology as source or target languages, if needed by mapping the language's alphabet to Latin characters.

Finally, while we intend to choose a diverse set of languages for this study, our overall number of languages is still rather small. This affects the generalizability of the results, and future work might want to look at larger samples of languages.

7 Related Work

Neural network models for inflection. Most research on inflectional morphology in NLP within the last years has been related to the SIGMORPHON and CoNLL–SIGMORPHON shared tasks on morphological inflection, which have been organized yearly since 2016 (Cotterell et al., 2016). Traditionally being focused on individual languages, the 2019 edition (McCarthy et al., 2019) contained a task which asked for transfer learning from a high-resource to a low-resource language. However, source–target pairs were predefined, and the question of how the source language influences learning besides the final accuracy score was not considered. Similarly to us, Gorman et al. (2019) performed a manual error analysis of morphological inflection systems for multiple languages. However, they did not investigate transfer learning, but focused on monolingual models.

Outside the scope of the shared tasks, Kann et al. (2017) investigated cross-lingual transfer for morphological inflection, but was limited to a quantitative analysis. Furthermore, that work experimented with a standard sequence-to-sequence model (Bahdanau et al., 2015) in a multi-task training fashion (Caruana, 1997), while we pretrain and fine-tune pointer–generator networks. Jin and Kann (2017) also investigated cross-lingual transfer in neural sequence-to-sequence models for morphological inflection. However, their experimental setup mimicked Kann et al. (2017), and the main research questions were different: While Jin and Kann (2017) asked how cross-lingual knowledge transfer works during multi-task training of neural sequence-to-sequence models on two languages, we investigate if neural inflection models demonstrate interesting differences in production errors depending on the pretraining language. Besides that, we differ in the artificial neural network architecture and language pairs we investigate.

Cross-lingual transfer in NLP. Cross-lingual transfer learning has been used for a large variety NLP of tasks, e.g., automatic speech recognition (Huang et al., 2013), entity recognition (Wang and Manning, 2014), language modeling (Tsvetkov et al., 2016), or parsing (Cohen et al., 2011; Søgaard, 2011; Ammar et al., 2016). Machine translation has been no exception (Zoph and Knight, 2016; Ha et al., 2016; Johnson et al., 2017). Recent research asked how to automatically select a suitable source language for a given target language (Lin et al., 2019). This is similar to our work in that our findings could potentially be leveraged to find good source languages.

Acquisition of morphological inflection. Finally, a lot of research has focused on human L1 and L2 acquisition of inflectional morphology (Salaberry, 2000; Herschensohn, 2001; Housen, 2002; Ionin and Wexler, 2002; Weerman et al., 2006; Zhang and Widyastuti, 2010).

To name some specific examples, Marqués-Pascual (2011) investigated the effect of a stay abroad on Spanish L2 acquisition, including learning of its verbal morphology in English speakers. Jia (2003) studied how Mandarin Chinese-speaking children learned the English plural morpheme. Nicoladis et al. (2012) studied the English past tense acquisition in Chinese–English and French–English bilingual children. They found that, while both groups showed similar production accuracy, they differed slightly in the type of errors they made. Also considering the effect of the native language explicitly, Yang and Huang (2004) investigated the acquisition of the tense-aspect system in an L2 for speakers of a native language which does not mark tense explicitly.

Finally, our work has been weakly motivated by Bliss (2006). There, the author asked a question for human subjects which is similar to the one we ask for neural models: How does the native language influence L2 acquisition of inflectional morphology?

8 Conclusion and Future Work

Motivated by the fact that, in humans, learning of a second language is influenced by a learner's native language, we investigated a similar question in artificial neural network models for morphological inflection: How does pretraining on different languages influence a model's learning of inflection in a target language?

We performed experiments on eight different source languages and three different target languages. An extensive error analysis of all final models showed that (i) for closely related source and target languages, acquisition of target language inflection gets easier; (ii) knowledge of a prefixing language makes learning of inflection in a suffixing language more challenging, as well as the other way around; and (iii) languages which exhibit an agglutinative morphology facilitate learning of inflection in a second language.

Future work might leverage those findings to improve neural network models for morphological inflection in low-resource languages, by choosing suitable source languages for pretraining.

Another interesting next step would be to investigate how the errors made by our models compare to those by human L2 learners with different native languages. If the exhibited patterns resemble each other, computational models could be used to predict errors a person will make, which, in turn, could be leveraged for further research or the development of educational material.

Acknowledgments

I would like to thank Samuel R. Bowman and Kyle Gorman for helpful discussions and suggestions. This work has benefited from the support of Samsung Research under the project *Improving Deep Learning using Latent Structure* and from the donation of a Titan V GPU by NVIDIA Corporation.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *TACL*, 4:431–444.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Heather Bliss. 2006. L2 acquisition of inflectional morphology: Phonological and morphological transfer effects. In *GASLA*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Shay B Cohen, Dipanjan Das, and Noah A Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *EMNLP*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *CoNLL–SIGMORPHON*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *SIGMORPHON*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL–HLT*.

Heidi C Dulay and Marina K Burt. 1974. Natural sequences in child second language acquisition 1. *Language learning*, 24(1):37–53.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *NAACL–HLT*.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Workshop on Evaluating Vector-Space Representations for NLP*.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. In *Workshop on Building Linguistically Generalizable NLP Systems*.

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. A linguist reads the output of the CoNLL-SIGMORPHON 2017 shared task. In *CoNLL*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv:1611.04798*.

Gunnar Ólafur Hansson. 2010. *Consonant harmony: Long-distance interactions in phonology*. University of California Press.

Julia Herschensohn. 2001. Missing inflection in second language French: accidental infinitives and other verbal deficits. *Second Language Research*, 17(3):273–305.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Alex Housen. 2002. A corpus-based study of the L2-acquisition of the English verb system. *Computer learner corpora, second language acquisition and foreign language teaching*, pages 77–116.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *IEEE*.

Tania Ionin and Kenneth Wexler. 2002. Why is is easier than -s?: acquisition of tense/agreement morphology by child second language learners of English. *Second Language Research*, 18(2):95–136.

Gisela Jia. 2003. The acquisition of the English plural morpheme by native Mandarin Chinese-speaking children. *JSLHR*, 46:1297–1311.

Huiming Jin and Katharina Kann. 2017. Exploring cross-lingual transfer of morphological knowledge in sequence-to-sequence models. In *SCLeM*.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. One-shot neural cross-lingual transfer for paradigm completion. In *ACL*.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *ACL*.

Katharina Kann and Hinrich Schütze. 2018. Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. In *EMNLP*.

Eric Kellerman. 1979. Transfer and non-transfer: Where we are now. *Studies in second language acquisition*, 2(1):37–57.

Aleksandr E Kibrik. 1998. Archi. *The handbook of morphology*, pages 455–476.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In **SEM*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting Pinker and Prince (1988) and the past tense debate. *TACL*, 6:651–665.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *ACL*.

Laura Marqués-Pascual. 2011. Study abroad, previous language experience, and Spanish L2 development. *Foreign Language Annals*, 44(3):565–582.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Miikka Silfverberg, Sebastian J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *SIGMORPHON*.

Elena Nicoladis, Jianhui Song, and Paula Marentette. 2012. Do young bilinguals acquire past tense morphology like monolinguals, only later? Evidence from French–English and Chinese–English bilinguals. *Applied Psycholinguistics*, 33(3):457–479.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Keren Rice. 2000. *Morpheme order and semantic scope: Word formation in the Athapaskan verb*. Cambridge University Press.

M Rafael Salaberry. 2000. The acquisition of english past tense in an instructional setting. *System*, 28(1):135–152.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 2018. IIT(BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *CoNLL–SIGMORPHON*.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *ACL–HLT*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *NAACL-HLT*.

Mengqiu Wang and Christopher D Manning. 2014. Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *TACL*, 2:55–66.

Fred Weerman, Jannetje Bisschop, and Laura Punt. 2006. L1 and L2 acquisition of Dutch adjectival inflection. *ACLC Working Papers*, 1(1):5–36.

Suying Yang and Yue-Yuan Huang. 2004. The impact of the absence of grammatical tense in L1 on the acquisition of the tense-aspect system in L2. *IRAL*, 42(1):49–70.

Yanyin Zhang and Ima Widyastuti. 2010. Acquisition of L2 English morphology. *Australian review of applied linguistics*, 33(3):29–1.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *NAACL-HLT*.

Modeling Conventionalization and Predictability within MWEs at the Brain Level

Shohini Bhattasali *
Dept. of Linguistics – UMIACS
University of Maryland
shohini@umd.edu

Murielle Popa–Fabre *
INRIA – University of Paris
ALMANACH – LLF
murielle.fabre@inria.fr

Christophe Pallier
Cognitive Neuroimaging Lab
INSERM-CEA
christophe@pallier.org

John Hale
Dept. of Linguistics
University of Georgia
jthale@uga.edu

Abstract

While expressions have traditionally been binarized as compositional and noncompositional in linguistic theory, Multiword Expressions (MWEs) demonstrate finer-grained distinctions. Using Association Measures like Pointwise Mutual Information and Dice's Coefficient, MWEs can be characterized as having different degrees of conventionalization and predictability. Our goal is to investigate how these gradiences could reflect cognitive processes. In this study, fMRI recordings of naturalistic narrative comprehension is used to probe to what extent these computational measures and the cognitive processes they could operationalize are observable during on-line sentence processing. Our results show that Dice's Coefficent, representing lexical predictability, is a better predictor of neural activation for processing MWEs. Overall our experimental approach demonstrates how we can test the cognitive plausibility of computational metrics by comparing it against neuroimaging data.

1 Introduction

Multiword Expressions (MWEs) are word clusters or expressions formed by more than a single word. Siyanova-Chanturia (2013) provides examples of MWEs in English to illustrate the wide variety among these expressions, as seen in Table 1. While they are a heterogenous family of expressions, what unifies them is a lack of compositional linguistic analysis and psycholinguistic evidence has been given for their predictability and conventionalization. Our unique approach is to adapt different computational metrics to describe the heterogeneity within these MWEs and whether it is observable at the brain level.

MWE comprehension was shown to be distinct from other kinds of language processing. For instance, it is well-established at the behavioral level that MWEs are produced and understood faster than matched control phrases due to their frequency, familiarity, and predictability (Siyanova-Chanturia and Martinez, 2014), in accordance with incremental processing from a psycholinguistic perspective (Clark and Wilkes-Gibbs, 1986; Clark and Marshall, 2002; Hale, 2006; Levy, 2008).This would follow if MWEs were remembered as chunks, in the sense of (Miller, 1956) that was later formalized by (Laird et al., 1986; Rosenbloom and Newell, 1987). In this study we investigate to what extent MWEs are processed as chunks or built-up compositionally during online sentence processing. By repurposing metrics which are traditionally used to identify collocations in corpus linguistics, we utilize them to investigate the different levels of compositionality within MWEs at the brain level.

Linguistic phenomena	Examples
fixed phrases	*per se*, *by and large*
noun compounds	*black coffee*, *cable car*
verb compounds	*give a presentation*, *come along*
binomials	*heaven and hell*, *safe and sound*
complex prepositions	*in spite of*
idioms	*break the ice*, *spill the beans*

Table 1: A wide variety of linguistic phenomena that are considered to be MWEs.

Earlier neuroimaging work on compositional-

*Co-first authors contributed equally to this work.

ity and lexical prediction by Willems et al. (2016) have addressed this issue in a broader sense using computational measures of entropy and surprisal. In natural language processing, MWEs have also been shown to have graded levels of compositionality (Salehi et al., 2015).

From a human language processing perspective, as Titone and Connine (1999) and Bhattasali et al. (2018) have discussed previously, these MWEs cannot simply be sorted into bipartite categories depending on whether they are processed as chunks or compositionally. Using the specific case of idioms, the authors in the first paper argue against an exclusively noncompositional or compositional approach and propose a hybrid approach to these expressions that ascribes non-compositional and compositional characteristics to these expressions. In a similar vein, the authors in the second paper provide neuroimaging evidence to show that these expressions fall along a graded spectrum and could be differentiated based on various aspects. Moreover, MWEs could be further distinguished based on predictability, modifiability, conventionalization, semantic opacity, among other aspects.

In this study, we utilize two Association Measures, Pointwise Mutual Information and Dice's Coefficient to capture respectively the degree of conventionalization and degree of predictability within these expressions. Furthermore, we probe whether these computational measures and their hypothesized cognitive instantiations are discernible at the cerebral level during naturalistic sentence processing.

2 Background

2.1 MWEs: A Gradient Approach

While Association Measures are commonly used in computational linguistics to identify MWEs since ngrams with higher scores are likely to be MWEs (Evert, 2008), in this study they are adapted as a gradient predictor to describe the MWEs within the text.

Krenn (2000) suggests that PMI and Dice are better-suited to identify high-frequency collocations whereas other association measures such as log-likelihood are better at detecting medium to low frequency collocations. Since MWEs are inherently high-frequency collocations (i.e., the words in an MWE tend to co-occur frequently with each other), these two association measures were chosen to describe the strength of association between the identified word clusters (cf. identification method in Al Saied et al. (2017)).

2.1.1 Pointwise Mutual Information

The first measure we use is Pointwise Mutual Information (PMI) (Church and Hanks, 1990). Intuitively, its value is high when the word sequence under consideration occurs more often together than one would have expected, based on the frequencies of the individual words (Manning et al., 1999). MWEs that receive a higher PMI score are seen as more conventionalized (Ramisch et al., 2010). Formally, PMI is a log-ratio of observed and expected counts:

$$\text{PMI} = log_2 \frac{c(w_n^1)}{E(w_n^1)} \tag{1}$$

2.1.2 Dice's Coefficient

The second measure used in this study is Dice's Coefficient (Dice, 1945; Sørensen, 1948). Dice's coefficient is used to identify rigid MWEs with strong association (Evert, 2008; Smadja et al., 1996). It is the ratio of the frequency of the sequence over the sum of the unigram frequency of the words in the sequence. E.g., for a bigram the two ratios are averaged by calculating their harmonic mean. The harmonic mean only assumes a value close to 1 (the largest possible Dice score) if there is a strong *prediction* in both directions, from `w1` to `w2` and vice versa. The association score will be much lower if the relation between the two words is asymmetrical.

This measure takes into account the length of the MWEs and the value ranges between 0 and 1:

$$\text{Dice} = \frac{n \times c(w_n^1)}{\Sigma_{i=1}^{n} c(w_i)} \tag{2}$$

A higher value for the Dice Coefficient indicates that the two tokens do not occur together by chance. While PMI is systematically higher at the end of a word cluster Dice is not. Since Dice coefficient focuses on cases of very strong association rather than the comparison with independence as PMI does, it can be interpreted as a measure of predictability (Evert, 2008). Moreover, compared to PMI, Dice coefficient captures words co-occurrence in a certain order.

2.2 Association Measures as a Cognitively Plausible Metric

While earlier work has focused on individual types of MWEs, this study investigates the cognitive processes underlying the comprehension of heterogeneous MWEs differing along the lexical association of the words that compose them. Specifically, it is hypothesized that different association measures would map onto different cognitive aspects of MWEs, such as how predictable they are, how cohesive they are, how conventionalized they are, how frozen they are etc.

MWE	PMI	Dice
boa constrictor	7.935	10
fairy tale	6.165	6.422
coloured pencil	6.545	1.926
heart skipped a beat	10	0.001
gesture of weariness	5.125	0.001
object of curiosity	5.096	0.001
a dirty trick	5.603	0.001
united states	1.859	0.005
against all odds	6.012	0.013
sense of urgency	6.255	0.004
christmas tree	4.485	1.233
good morning	3.783	1.433
find out	3.479	1.240
come into	3.067	0.683

Table 2: Example of MWEs with two Association Measures: Pointwise Mutual Information and Dice's Coeffecient. Values highlighted in dark green indicate high scores while values highlighted in light green indicate low scores.

Thus, these association measures are used and adapted to describe different facets of MWEs. As presented above, PMI is taken to quantify the degree of conventionalization within these MWEs (Ramisch et al., 2010). Dice is taken to represent the degree of predictability of these MWEs (Evert, 2008). In Table 2, we can compare these measures on a set of identified word clusters. For example, expressions like *object of curiosity*, *gesture of weariness*, and *heart skipped a beat* would be considered highly conventionalized given their high PMI score but less predictable, given their low Dice score. As per these metrics, both *boa constrictor* and *fairy tales* are highly conventionalized and highly predictable whereas expressions like *united states* and *come into* are neither highly conventionalized nor highly predictable.

If we visually compare these scores for all 669 unique MWEs, as in Figure 1 below, we can also notice an interesting pattern. The values for PMI are spread across the axis and thus, the expressions are along a graded spectrum of conventionalized and have more fine-grained distinctions. On the other hand, since Dice is used to identify rigid MWEs, it tends to cluster the expressions around each end of the spectrum. We interpret these two different distributions of variance as enabling us to model different cerebral activation patterns of lexical association in MWEs processing at the brain level. Thus we repurpose Dice and PMI to represent different ongoing lexical processes.

Wiechmann (2008) also gave a cognitive dimension to the idea of association measures in order to investigate the association between a verb and its syntactic frames. He evaluated the measures against how well it could predict human reading behavior in an eye-tracking study. Our approach is similar to Wiechmann's cognitive-oriented approach since we also compare different association measures and test it against neural data, instead of behavioral data. An earlier study by Bhattasali et al. (2018) has illustrated how PMI specifically can be used to show not only the graded spectrum of compositionality within MWEs, but also how the more cohesive expressions implicate memory-related areas whereas the less cohesive expressions implicate well-known syntactic structure-building areas.

3 fMRI Study

3.1 Method

Participants hear the story over headphones while they are in the scanner. The sequence of neuroimages collected during their session becomes the dependent variable in a regression against word-by-word predictors, derived from the text of the story (cf. Table 3).

3.2 Stimuli & MWE Identification

The English audio stimulus was Antoine de Saint-Exupéry's *The Little Prince*, translated by David Wilkinson and read by Nadine Eckert-Boulet. It constitutes a fairly lengthy exposure to naturalistic language, comprising 19,171 tokens; 15,388

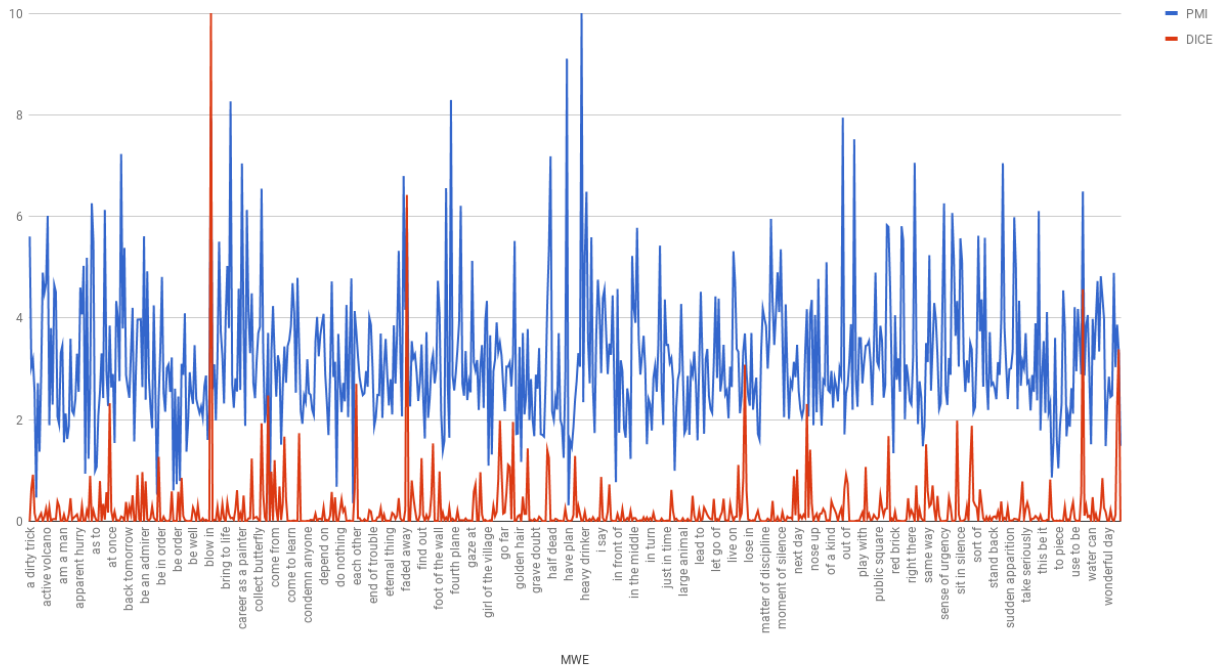


Figure 1: Comparing Pointwise Mutual Information (in blue) with Dice's Coefficient (in red); the former illustrates more fine-grained gradience; scaled up for visual purposes

words and 1,388 sentences, and lasting over an hour and a half.

Within this text, 669 MWEs were identified using a transition-based MWE analyzer (Al Saied et al., 2017). Al Saied et al. use unigram and bigram features, word forms, POS tags and lemmas, in addition to features such as transition history and report an average F-score 0.524 for this analyzer across 18 different languages which reflects robust cross-linguistic performance. The analyzer was trained on examples from the Children's Book Test (CBT) from the Facebook bAbI project (Hill et al., 2015) to keep the genre consistent with our literary stimulus. This corpus consists of text passages that are drawn from the Children's section of Project Gutenberg, a free online text repository. External lexicons were also used to supplement the MWEs found with the analyzer. The external lexicons included the Unitex lexicon (Paumier et al., 2009), the SAID corpus (Kuiper et al., 2003), the Cambridge International Dictionary of Idioms (White, 1998), and the Dictionary of American Idioms (Makkai et al., 1995).

3.3 Participants

56 participants were scanned and 5 of them were excluded since they had incomplete scanning sessions. Participants included were fifty-one volunteers (32 women and 19 men, 18-37 years old) with no history of psychiatric, neurological, or other medical illness or history of drug or alcohol abuse that might compromise cognitive functions. All strictly qualified as right-handed on the Edinburgh handedness inventory (Oldfield, 1971). They self-identified as native English speakers and gave their written informed consent prior to participation, in accordance with Cornell University IRB guidelines.

3.4 Presentation

Participants listened to the entire audiobook for 1 hour and 38 minutes. The story had nine chapters and at the end of each chapter the participants were presented with a multiple-choice questionnaire with four questions (36 questions in total), concerning events and situations described in the story. These questions served to confirm participants' comprehension. They were viewed via a mirror attached to the head coil and answered through a button box. The entire session lasted around 2.5 hours.

3.5 Data Collection

Imaging was performed using a 3T MRI scanner (Discovery MR750, GE Healthcare, Milwaukee, WI) with a 32-channel head coil at the Cornell MRI Facility. Blood Oxygen Level Dependent (BOLD) signals were collected using a T2

-weighted echo planar imaging (EPI) sequence (repetition time: 2000 ms, echo time: 27 ms, flip angle: 77deg, image acceleration: 2X, field of view: 216 x 216 mm, matrix size 72 x 72, and 44 oblique slices, yielding 3 mm isotropic voxels). Anatomical images were collected with a high resolution T1-weighted (1 x 1 x 1 mm^3 voxel) with a Magnetization-Prepared RApid Gradient-Echo (MP-RAGE) pulse sequence.

4 Data Analysis

4.1 Preprocessing

fMRI data is acquired with physical, biological constraints and preprocessing allows us to make adjustments to improve the signal to noise ratio. Primary preprocessing steps were carried out in AFNI version 16 (Cox, 1996) and include motion correction, coregistration, and normalization to standard MNI space. After the previous steps were completed, ME-ICA (Kundu et al., 2012) was used to further preprocess the data. ME-ICA is a denoising method which uses Independent Components Analysis to split the T2*-signal into BOLD and non-BOLD components. Removing the non-BOLD components mitigates noise due to motion, physiology, and scanner artifacts (Kundu et al., 2017).

4.2 Statistical Analysis

The research questions presented above in section 2 motivates a statistical analysis that performs a comparison where fMRI signal is modeled in two General Linear Models (GLM) : one by Dice scores tagged on the identified MWEs (Model 2) versus one where PMI scores are quantifying the conventionality of each MWE in the Little Prince (Model 1).

fMRI data were analyzed in the following way: for each subject, and at each brain location (voxel), the time course of activation was submitted to a multiple linear regression that estimated the specific effect of each predictor (cf. 4.2.1), after convolution by a standard hemodynamic response (Poldrack et al., 2011).

The effects of the predictors - the increase in r^2 associated to them - were then submitted to second level analyses to test for significance at the group level. Model comparisons using root-means square (r^2) maps was carried out using a Python pipeline in order to evaluate the goodness of fit of the two Association Measures with BOLD signal (cf. 4.2.2).

4.2.1 GLM Analyses: Single-subject statistics

At the single-subject level, the observed timecourse of the brain's hemodynamic response (BOLD - Blood Oxygenation Level Dependent) in each voxel was modeled by the predictors in Table 3 including one of the two Association Measures under analysis calculated as illustrated in formulas given in 2.1), and time-locked at the offset of each word or MWE in the audio-book*.

The predictors shown in Table 3 were convolved using SPM's canonical HRF (Hemodynamic Response Function, Friston et al. (2007)). The two neuroimaging models (i.e. with PMI or with Dice) also included four control variables (confounds) as shown in Table 3.

Model 1: with PMI We regressed the word-by-word predictors described below against fMRI timecourses recorded during passive story-listening in a whole-brain analysis. For each of the 15,388 words in the story, their timestamps were estimated using Praat TextGrids (Boersma, 2002). MWEs were identified, as described in §3.2 and all 669 unique MWEs were annotated with their PMI score. This score is based on corpus frequency counts from the Corpus of Contemporary English (Davies, 2008), and were calculated using `mwetoolkit` (Ramisch et al., 2010; Ramisch, 2012) and the formula given above in 2.1. COCA is a large, genre-balanced corpus of American English and contains contains more than 560 million words of text, equally divided among spoken, fiction, popular magazines, newspapers, and academic texts.

Additionally, we entered four regressors of non-interest into the regression analysis: word offset, word frequency (Brysbaert and New, 2009), pitch, intensity which serve to improve the sensitivity, specificity and validity of activation maps (Bullmore et al., 1999; Lund et al., 2006). These predictors were added to ensure that conclusions about MWE processing would be specific to the cognitive processes they were taken to instantiate, as opposed to more general aspects of speech perception. Specifically, lexical frequency of each word was added as a covariate of non-interest, to statistically factor out effects of general word frequency, that may correlate with other types of

*For more details about the hemodynamic response, please see chapter 2 of Kemmerer (2014).

Predictors	Description
Association Measure PMI or DICE	Word-by-word on MWEs (§2.1)
Word rate	Tags the offset of each spoken word in time
Word frequency	Word-by-word log-frequency in movie subtitles
F0	Fundamental frequency of the narrator's voice, which reflects pitch
RMS amplitude	Root Mean Square Amplitude of the narrators voice, which reflects intensity, an acoustic correlate of volume

Table 3: Predictors used in the fMRI Analysis.

expectations. To control for sentence-level and phrase-level compositional processes, we included a regressor formalizing syntactic structure building based on a bottom-up parsing algorithm (Hale, 2014), as determined by the Stanford parser (Klein and Manning, 2003). Controlling for structural composition allows us to isolate and focus our investigation on noncompositional processing, as in MWEs. These regressors were not orthogonalized.

Model 2: with Dice Model 2 is similar to Model 1 and uses the same predictors. However, instead of PMI scores, the MWEs were annotated with their corresponding Dice's coefficient scores. These were also calculated using corpus frequency counts from COCA and the `mwetoolkit`.

4.2.2 r^2 Model comparison

The research questions presented above in section 2 motivates a statistical analysis that performs a comparison where fMRI signal on MWEs is modeled in the above presented GLMs by PMI versus Dice measures.

r^2 model comparison For every subject, we compute how much the inclusion of each variable of interest (i.e. Dice and PMI) increases the cross-validated r^2. Hence, the r^2 scores represent the variance explained in each voxel by the variable instantiating the MWE processing Dice or PMI respectively provide.

Group-level statistics To compare the impact of the two variables on fMRI signal explanation (i.e. r^2 increase of each variable), we performed a paired t-test on each individual r^2 brain map, and obtained the map in Figure 2 showing where one of the variables explains significantly better the signal than the other (see clusters on Table 4).

5 Results - Fit with fMRI signal

We performed an r^2 comparison to test which Association Measure on MWEs provided the better fit to the fMRI signal recorded during *The Little Prince*.

Dice vs. PMI The two different Association Measure were tested (Dice and PMI), and Dice, taken to represent the degree of predictability, was shown to be the best fitting the BOLD signal of these two models. Figure 2 (clusters coordinates and statistics, cf. Table 4), shows the significance (z-scores after Bonferroni correction with $p < 0.05$) of the difference in r^2 scores with a cluster threshold of 10 voxels.

Of the two Association Measures , the Dice measure (i.e. degree of predictability) had a significant predictive value in well-known language areas such as temporal regions, although mainly right-lateralized.

6 Discussion

The present neuroimaging study offers a first experimental grounding to the fact that a computational measure instantiating lexical prediction has a better fit with brain activity elicited by processing MWEs in certain regions of the language network. In both anterior and posterior portions of language network - and specifically in temporal areas - this lexical knowledge based process has a significant predictive value.

This result is in line with earlier work on lexical prediction with computational measures like entropy and surprisal by Willems et al. (2016) where temporal regions were identified together with right lateralized frontal ones.

Assuming Dice operationalizes some predictive processes within complex lexical items, these predictive processes are plausibly linked to higher demands in semantic combinatorial operations, as

Regions for Dice >PMI	Cluster size (in voxels)	MNI Coordinates x	y	z	z-scores
R Superior Temporal Gyrus (BA 38)	47	48	10	-26	5.80
R Middle Temporal Gyrus	84	54	-18	-10	6.09
R Middle Temporal Gyrus (BA 22)	98	48	-36	2	5.85
R Superior Temporal Gyrus (BA 22)	70	48	-12	2	5.83
R Middle Temporal Gyrus (BA 22)	16	58	-46	2	5.14
L Superior Temporal Gyrus	13	-62	-18	6	5.64
R Superior Frontal Gyrus	10	20	56	12	5.53
R Inferior Frontal Gyrus (BA 45)	10	48	20	14	5.64
L Supramarginal Gyrus	22	-56	-56	22	5.37
R Inferior Parietal Lobule/ Superior Temporal Gyrus (BA 40)	10	62	-46	22	5.44
R Inferior Parietal Lobule/ Superior Temporal Gyrus (BA 40)	16	54	-46	22	5.45
R Superior Frontal Gyrus	35	20	42	34	5.69
R Cingulate Gyrus	17	2	-34	34	5.85
R Precenus	22	32	-72	36	5.76
L Inferior Parietal Lobule	12	-34	-58	46	5.17

Table 4: Significant clusters for Dice's Coefficient versus Pointwise Mutual Information after Bonferroni correction with p < 0.05, based on R2 analysis in §4.2.2, and shown in Figure 2

.

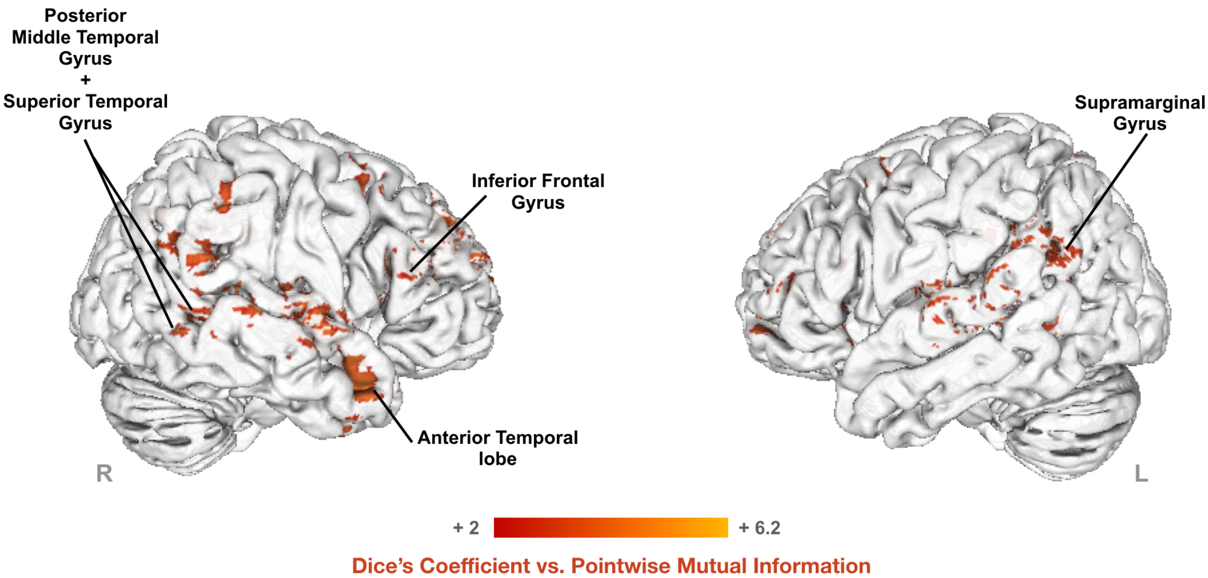


Figure 2: Z-map showing regions having a significant effect for Dice's coefficient versus Pointwise Mutual Information after Bonferroni correction with p < 0.05

reported in previous neuroimaging studies investigating semantic combinatorial processes through comparing meaningful and less meaningful word combinations (Price et al., 2015; Graves et al., 2010). Crucially, the graded psycholinguistic measures about lexical combination tested in these studies elicit similar areas as the regions where a better fit to the fMRI signal is observed in the present study.

Based on the formula, Dice helped us to factor out effects of length in longer MWEs and provided us with a more abstract measure given its bidirectional association. This could be a reason that it was a better fit to the BOLD signal, compared to PMI which is biased based on the length of the expression.

Lastly, Dice's Coefficient is a more rigid measure of lexical association compared to Pointwise Mutual Information, as seen in Fig. 1. Hence, Dice clusters highly predictable expressions versus less predictable ones, giving rise to two main groups. PMI displays more fine-grained distinctions overall (compared to Dice) and thus, captures the spectrum of compositional gradience within these MWEs as shown in a previous neuroimaging study. Bhattasali et al. (2018) showed that increasing values of PMI activates the network of syntactic building. However, the fact that Dice is the better fit between the two is interesting since it suggests that a bimodal distribution of gradience is cognitively more plausible than a fine-tuned approach to gradience, specifically in posterior temporal areas. Thus, this paves the way for further investigations regarding which computational measures are more cognitively pertinent to grasp a better understanding of human cognition and its neural substrates.

7 Conclusion & Further Work

Overall, this study examines MWEs through the lens of two different Association Measures, Pointwise Mutual Information and Dice's Coefficent. We investigate to what extent these computational measures, operationalizing conventionalization and predictability, and their underlying cognitive processes are observable during on-line sentence processing. Our results show that Dice's Coefficient, formalizing the degree of predictability, is a better predictor of cerebral activation for processing MWEs and this suggests it is a more cognitively plausible computational metric in temporal areas where previous neuroimaging literature identified lexical predictive processes.

Apart from Association Measures, a future approach would be to investigate different metrics to capture other nuances between these MWEs. There are alternate approaches to describes MWEs such as word space models, based on distributional semantics, which could also serve as a metric of compositionality for these noncompositional word clusters. This type of metric would utilize the distributional patterns of words collected over large text data to represent semantic similarity between words in terms of spatial proximity (Sahlgren, 2006).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1607441 and by the French National Research Agency (ANR) under grant ANR-14-CERA-0001. This research has also been supported in part by ONR MURI Award N00014–18–1–2670.

We are grateful to Wenming Luh for help in preprocessing, to Hazem Al Saied and Matthieu Constant for help in identifying MWEs, and to Eric de la Clergerie for valuable help in interpreting Association Measures. We also thank the anonymous reviewers for their detailed comments on the original manuscript.

References

Hazem Al Saied, Marie Candito, and Matthieu Constant. 2017. The ATILF-LLF system for the PARSEME Shared Task: a Transition-based Verbal Multiword Expression Tagger. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 127–132, Valencia, Spain. Association for Computational Linguistics.

Shohini Bhattasali, Murielle Fabre, and John Hale. 2018. Processing MWEs: Neurocognitive bases of verbal MWEs and lexical cohesiveness within MWEs. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 6–17.

Paul Boersma. 2002. *Praat, a system for doing phonetics by computer*. Glot International.

Marc Brysbaert and Boris New. 2009. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior research methods*, 41(4):977–990.

ET Bullmore, MJ Brammer, S Rabe-Hesketh, VA Curtis, RG Morris, SCR Williams, T Sharma, and PK McGuire. 1999. Methods for diagnosis and treatment of stimulus-correlated motion in generic brain activation studies using fMRI. *Human brain mapping*, 7(1):38–48.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Herbert H Clark and Catherine R Marshall. 2002. Definite reference and mutual knowledge. *Psycholinguistics: critical concepts in psychology*, 414.

Herbert H Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Robert W. Cox. 1996. Afni: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical research*, 29(3):162–173.

Mark Davies. 2008. *The Corpus of Contemporary American English (COCA): 560 million words, 1990–present*. BYE, Brigham Young University.

Lee R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Stefan Evert. 2008. Corpora and collocations. *Corpus linguistics. An international handbook*, 2:1212–1248.

K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny, editors. 2007. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press.

William W Graves, Jeffrey R Binder, Rutvik H Desai, Lisa L Conant, and Mark S Seidenberg. 2010. Neural correlates of implicit and explicit combinatorial semantic processing. *Neuroimage*, 53(2):638–646.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):643–672.

John T Hale. 2014. *Automaton theories of human sentence comprehension*. CSLI Publications.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

David Kemmerer. 2014. *Cognitive neuroscience of language*. Psychology Press.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Brigitte Krenn. 2000. Empirical implications on lexical association measures. In *Proceedings of The Ninth EURALEX International Congress*.

Koenraad Kuiper, Heather McCann, Heidi Quinn, Therese Aitchison, and Kees van der Veer. 2003. Syntactically Annotated Idiom Dataset (SAID) LDC2003T10. In *Linguistic Data Consortium*, Philadelphia.

Prantik Kundu, Souheil J Inati, Jennifer W Evans, Wen-Ming Luh, and Peter A Bandettini. 2012. Differentiating bold and non-bold signals in fMRI time series using multi-echo epi. *Neuroimage*, 60(3):1759–1770.

Prantik Kundu, Valerie Voon, Priti Balchandani, Michael V. Lombardo, Benedikt A. Poser, and Peter A. Bandettini. 2017. Multi-echo fMRI: A review of applications in fMRI denoising and analysis of bold signals. *NeuroImage*, 154:59 – 80.

John Laird, Paul Rosenbloom, and Allen Newell. 1986. Chunking in Soar, anatomy of a general learning mechanism. *Machine Learning*, 1.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Torben E Lund, Kristoffer H Madsen, Karam Sidaros, Wen-Lin Luo, and Thomas E Nichols. 2006. Non-white noise in fMRI: does modelling have an impact? *Neuroimage*, 29(1):54–66.

Adam Makkai, M. T. Boatner, and J. E. Gates. 1995. *A Dictionary of American idioms*. ERIC.

Christopher D Manning, Hinrich Schütze, et al. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

Richard C Oldfield. 1971. The assessment and analysis of handedness: the Edinburgh inventory. *Neuropsychologia*, 9(1):97–113.

Sébastien Paumier, Takuya Nakamura, and Stavroula Voyatzi. 2009. Unitex, a corpus processing system with multi-lingual linguistic resources. *eLEX2009*, page 173.

Russell A Poldrack, Jeanette A Mumford, and Thomas E Nichols. 2011. *Handbook of functional MRI data analysis*. Cambridge University Press.

Amy R Price, Michael F Bonner, Jonathan E Peelle, and Murray Grossman. 2015. Converging evidence for the neuroanatomic basis of combinatorial semantics in the angular gyrus. *Journal of Neuroscience*, 35(7):3276–3284.

Carlos Ramisch. 2012. A generic framework for multiword expressions treatment: From acquisition to applications. In *Proceedings of ACL 2012 Student Research Workshop*, pages 61–66. Association for Computational Linguistics.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. mwetoolkit: a Framework for Multiword Expression Identification. In *LREC*, volume 10, pages 662–669.

Paul S. Rosenbloom and Allen Newell. 1987. Learning by chunking: A production-system model of practice. In *Production System Models of Learning and Development*, pages 221–286. MIT Press.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983.

Anna Siyanova-Chanturia. 2013. Eye-tracking and ERPs in multi-word expression research: A state-of-the-art review of the method and findings. *The Mental Lexicon*, 8(2):245–268.

Anna Siyanova-Chanturia and Ron Martinez. 2014. The idiom principle revisited. *Applied Linguistics*, 36(5):549–569.

Frank Smadja, Kathleen R McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: A statistical approach. *Computational linguistics*, 22(1):1–38.

Thorvald Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34.

Debra A Titone and Cynthia M Connine. 1999. On the compositional and noncompositional nature of idiomatic expressions. *Journal of pragmatics*, 31(12):1655–1674.

Daniel Wiechmann. 2008. On the computation of collostruction strength: Testing measures of association as expressions of lexical bias. *Corpus Linguistics and Linguistic Theory*, 4(2):253–290.

Roel M Willems, Stefan L Frank, Annabel D Nijhof, Peter Hagoort, and Antal Van den Bosch. 2016. Prediction during natural language comprehension. *Cerebral Cortex*, 26(6):2506–2516.

DialectGram: Detecting Dialectal Variation at Multiple Geographic Resolutions

Hang Jiang*
Symbolic Systems
hjian42@stanford.edu

Haoshen Hong*
Computer Science
haoshen@stanford.edu

Yuxing Chen*
Symbolic Systems
yxchen28@stanford.edu

Vivek Kulkarni
Computer Science
viveksck@stanford.edu

Abstract

Several computational models have been developed to detect and analyze dialect variation in recent years. Most of these models assume a predefined set of geographical regions over which they detect and analyze dialectal variation. However, dialect variation occurs at multiple levels of geographic resolution ranging from cities within a state, states within a country, and between countries across continents. In this work, we propose a model that enables detection of dialectal variation at multiple levels of geographic resolution obviating the need for a-priori definition of the resolution level. Our method DIALECTGRAM, learns dialect-sensitive word embeddings while being agnostic of the geographic resolution. Specifically it only requires one-time training and enables analysis of dialectal variation at a chosen resolution post-hoc – a significant departure from prior models which need to be re-trained whenever the pre-defined set of regions changes. Furthermore, DIALECTGRAM explicitly models senses thus enabling one to estimate the proportion of each sense usage in any given region. Finally, we quantitatively evaluate our model against other baselines on a new evaluation dataset `DialectSim` (in English) and show that DIALECTGRAM can effectively model linguistic variation.

1 Introduction

Studying regional variation of language is central to the field of sociolinguistics. Traditional approaches (Labov, 1980; Milroy, 1992; Tagliamonte, 2006; Wolfram and Schilling, 2015) focus on rigorous manual analysis of linguistic data collected through time-consuming and expensive surveys and questionnaires. The evolution of the Internet and social media now enables studying linguistic variation at a scale thus overcoming some of the scalability challenges faced by survey based methods. Consequently, computational methods to detect and analyze geographic variation in language have been proposed (Eisenstein et al., 2010, 2011, 2014; Bamman et al., 2014; Kulkarni et al., 2015b)

However, most prior work suffers from three limitations: First, previous models (Kulkarni et al., 2015b) such as Frequency Model, Syntactic Model, and `GEODIST` all rely on pre-defined regional classes to model linguistic changes (an exception is (Eisenstein et al., 2010) which focuses on lexical variation). The use of pre-defined regional classes limits the flexibility of these baseline models because dialect changes can be observed at various geographic resolutions. Second, previous models do not explicitly model the sense distribution of each word. In this work, we address these limitations by proposing a model DIALECTGRAM that enables analysis at multiple geographic resolutions while explicitly modeling word senses (see Figures 1 - 4). Given a corpus which can be associated with geographical regions, DialectGram first induces the number of senses for each word using a non-parametric Bayesian model (Bartunov et al., 2016). This step requires no apriori knowledge of the geographic resolution[1]. Having inferred the senses of each word, we show how to detect and analyze dialectal variation at any chosen geographic resolution by clustering usages in any given region based on their sense usage.

To summarize, our contributions are:

- **Multi-resolution Model:** We introduce DIALECTGRAM, a method to study the geographic variation in language across multiple

*Equal contribution.

[1]The only requirement is that the corpus be geo-tagged so that analysis can be conducted post-hoc at any desired resolution.

levels of resolution without assuming knowledge of the geographical resolution apriori.

- **Explicit Sense modeling:** DIALECTGRAM predicts how likely each sense of a word is used in a context thus enabling a more precise modeling of linguistic change.
- **Corpus and Validation Set:** We build a new English Twitter corpus `Geo-Tweets2019` for training dialect-sensitive word embeddings. Furthermore, we construct a new validation set `DialectSim` for evaluating the quality of English region-specific word embeddings between UK and USA.

2 Related Work

Linguistic variation. In the past, sociologists and linguists have been studying linguistic change by designing experiments to manually collect data (Labov, 1980; Milroy, 1992) and conducting variation analysis (Tagliamonte, 2006). Several works (Eisenstein et al., 2010; Gulordava and Baroni, 2011; Kim et al., 2014; Jatowt and Duh, 2014; Kulkarni et al., 2015a,b; Kenter et al., 2015; Gonçalves and Sánchez, 2016; Donoso and Sanchez, 2017; Lucy and Mendelsohn, 2018; Shoemark et al., 2019) have used different computational models to study dialect variations with respective to geography, gender, and time.

Eisenstein et al. (2010) is one of the first to tackle the linguistic variation problem with computational models. They design a multi-level generative model that uses latent topic and geographic variables to analyze lexical variation in English. This latent variable model is able to generate an author's geographic location based on the author's text. To quantitatively evaluate the models, they compute the physical distance between the prediction and the true location. Similarly, Gonçalves and Sánchez (2016) apply K-means method to cluster the geographic lexical superdialects assuming a list of pre-defined set of words that are known to demonstrate lexical variation. This was followed by Gonçalves and Sánchez (2016) who propose two metrics to calculate the linguistic distance between geographic regions. That is, instead of using the physical distance between the predicted and the true location, they compute cosine similarities or Jensen-Shannon Divergence (JSD) to evaluate the model quantitatively.

Recently, Kulkarni et al. (2015b) building on the work of (Bamman et al., 2014) propose a word embeddings based model `GEODIST` model for robustly modeling dialectal variation and focuses on capturing semantic changes between dialects. Nevertheless, a pre-defined set of regions is required for the model to update region-specific embeddings. For instance, Kulkarni et al. (2015b) assume that English exhibits dialectal variation between the US and UK, and train the network to learn two sets of word embeddings for the two regions. However, a model trained using this data cannot be used to analyze dialectal variation across states or any other level of resolution without a re-training from scratch. To learn how English changes within each state, Kulkarni et al. (2015b) would need to tag each US tweet with a state name and train the model again. Moreover, the model does not explicitly capture senses of a word but only learns region specific embeddings.

Word Sense Disambiguation. The problem of detecting dialectal variants of a word can be viewed broadly in terms of word sense induction where the different word senses can roughly correspond to usages in different regions. For instance, the word *pants* usually refer to *underwear* in the US versus *trousers* in the UK, suggesting two senses for *pants*. Consequently, we discuss the most relevant work on word sense induction as well. Reisinger and Mooney (2010) is the first paper that modifies the single *prototype* vector space model to obtain multi-sense word embeddings with average cluster vectors as prototypes. Many works (Huang et al., 2012; Neelakantan et al., 2014; Tian et al., 2014; Chen et al., 2014) are later dedicated to combine Skip-gram, clustering algorithm, and linguistic knowledge to learn word senses and embeddings jointly. Bartunov et al. (2016) adopt a non-parametric Bayesian approach and propose the Adaptive Skip-gram (AdaGram) model, which is able to induce word senses without assuming any fixed number of prototypes. As we will see in the following sections, we build on precisely this approach to model regional variation.

3 Data

3.1 Geo-Tweets2019 Corpus

We create a new corpus, `Geo-Tweets2019`, which consists of English tweets[2] during April and May in 2019 from the United States and the United Kingdom. Each tweet includes the user ID, the

[2]We use the Tweepy toolkit.

Word	US Meaning	UK Meaning
flat	smooth and even; without marked lumps or indentations	apartment
flyover	flypast, ceremonial aircraft flight	elevated road section
pants	trousers	underwear
lift	elevator	raise
football	soccer	American football

Table 1: Examples of words that have different meanings in American and British English

published time, the geographic location, and tweet text. We have around 2M tweets from the US and 1M from the UK. We preprocessed the tweets with the tweet tokenizer from Eisenstein et al., 2010 and regular expressions. Finally, we filtered out URL's, emojis, and other irregular uses of English to shrink the size of vocabulary and to facilitate the training of word vectors. Statistics can be seen in Table 2.

Number	US	UK	Total
tweet	2,075,394	1,088,232	3,163,626
token	41,637,107	22,012,953	63,650,060
term	865,784	469,570	1,167,790

Table 2: Statistics of Geo-Tweets2019

3.2 DialectSim Validation Set

To evaluate the models, we construct a new validation set `DialectSim`, which comprises of words with same or shifted meanings in the US and the UK. To build this validation set, we first crawled a list of words that show different meanings from the Wikipedia page[3] and pick 341 words that appear more than 20 times in our corpus in the UK and the US. Table 1 presents three examples in the dataset. In order to generate balanced positive and negative samples, we sample another 341 negative examples randomly from our `Geo-Tweets2019` dataset. A minimum frequency of 20 is also used for negative sampling. These negative cases were manually verified by each of the three authors independently. Finally, we split the dataset into training set with 511 samples (75%) and testing set with 171 samples (25%).

4 Models

4.1 Baseline Models

Frequency Model. One baseline method to detect whether there are significant changes between usage in two regions is to count the occurrence of a word in the US and the UK tweets. We have implemented this Frequency Model as described in Kulkarni et al. (2015b).

Syntactic Model. A more nuances approach compared to the frequency based approach is to detect change in syntactical roles across regions. The Syntactic Model (Kulkarni et al., 2015b) takes Part-of-Speech (POS) tag into consideration as well. More specifically, if a word is used equally frequently in both countries, but the their POS usages are different, then we consider the meaning of two words as different between two countries. We use the CMU ARK Twitter Part-of-Speech Tagger[4] for POS tagging.

GEODIST (Skip-gram) Model. The main idea of `GEODIST` model (which can detect semantic changes) (Kulkarni et al., 2015b) is to learn region-specific word embeddings and use boot-strapping to estimate confidence scores on detected changes. Instead of learning a single vector to represent a word, this model aims to jointly learn a global embedding $\delta_{\text{MAIN}}(w)$ as well as (multiple) differential embeddings $\delta_{r_i}(w)$ for each word w in the vocabulary with $R = (r_1, r_2, \ldots)$ geographical regions exactly as described in (Bamman et al., 2014). In particular, the region-specific embedding is defined as the sum of the global embedding and the differential embedding for that region: $\phi_{r_i}(w) = \delta_{\text{MAIN}(w)} + \delta_{r_i}(w)$. The objective function is to minimize the negative log-likelihood of the context word given the center word conditioned on the region. We use stochastic gradient descent method (Bottou, 1991) to update the model parameters. We implement our own `GEODIST` model in PyTorch.

4.2 DialectGram Model

We construct a new model for detecting dialectal changes which we called DIALECTGRAM (Dialectal Adaptive Skip-gram). The model first learns multi-sense word embeddings using Ada-

[3] https://en.wikipedia.org/wiki/Lists_of_words_having_different_meanings_in_American_and_British_English

[4] http://www.cs.cmu.edu/~ark/TweetNLP/

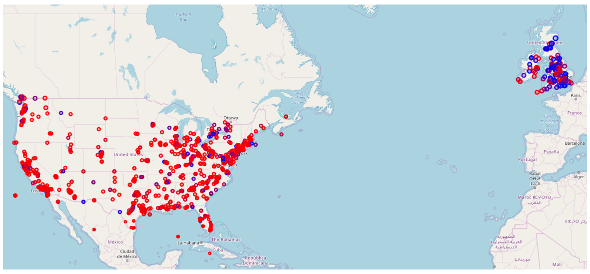


Figure 1: Dialectal variation of *gas* across countries. Tweets that contain *gas* with predicted sense "gaseous substance" are illustrated as blue circles; tweets that contain *gas* with predicted sense "gasoline" are plotted as red circles.

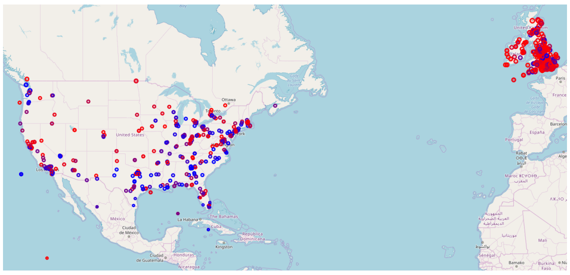


Figure 2: Dialectal variation of *flat* across countries. Tweets that contain *flat* with predicted sense "apartment" are illustrated as red circles; tweets that contain *flat* with predicted sense "smooth and even" are plotted as blue circles.

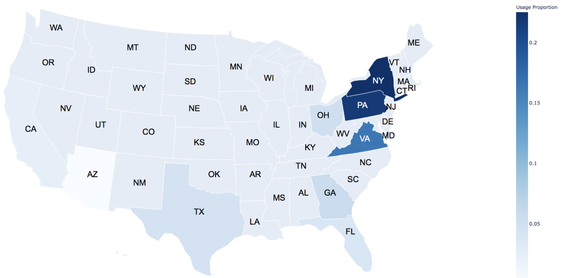


Figure 3: Dialectal variation of *buffalo* across US states. Here we show for each state, the proportion of sense 1 usage (*Buffalo city*) in blue. Grey indicates that the state contains no tweet using the word *buffalo* in our corpus.

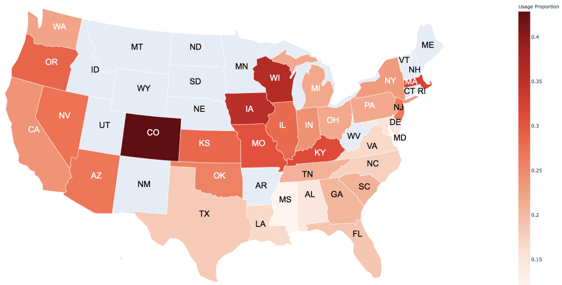


Figure 4: Dialectal variation of *pop* across US states. Here we show for each state, the proportion of sense 2 usage (*soft drink, soda*) in red. Grey indicates that the state contains no tweet using the word *pop* in our corpus.

gram (Bartunov et al., 2016) through training on the region-agnostic corpus. Once sense specific embeddings are obtained, based on the chosen resolution the model composes region-specific word embeddings by taking a weighted average of sense embeddings. At last, the model calculates the distance between region-specific word embeddings of the same word to determine whether a significant change exists. Our method is described succinctly in Algorithm 1.

Compared to the GEODIST model which needs predefined geographic label to update the region-specific embeddings, DIALECTGRAM learns multi-sense word embeddings on our dataset without any knowledge of the underlying regions.For instance, DialectGram automatically induces and learns the two senses of the word *flat* which could mean an *apartment* or *level land* corresponding to usages in the UK and US respectively.

Implementation details We train our model on our Geo-Tweets2019 corpus to learn word sense embeddings using the Julia implementation of

Algorithm 1 Use DIALECTGRAM to Compose Region-specific Embeddings

Input: w word
Output: e_r weighted region embedding for w

Load the trained DIALECTGRAM model
Build $Index_r$ on $Corpus$ from region r
for $s, p \in \text{GETSENSEPRIORS}(w)$ **do**
  $S_c[s] \leftarrow 0$, $S_p[s] \leftarrow p$ ▷ Note: S_c is sense counts, S_p is sense priors
end for
for all $c \in \text{GETCONTEXTS}(w)$ **do**
  $s \leftarrow \text{DISAMBIGUATE}(w, c)$
  $S_c[s] \leftarrow S_c[s] + 1$
end for
$e_r \leftarrow \text{GETWEIGHTEDVECTOR}(S_c, S_p)$

AdaGram[5] and then implement the inference algorithm in Python. To obtain a word's region-specific embedding in a place, we first use DIALECTGRAM to predict the dominant sense for the word in each tweet from a region and use weighted average of the sense embeddings as the region-specific word embedding e_r. We use the fol-

[5] https://github.com/sbos/AdaGram.jl

lowing hyper-parameter settings: $\texttt{min\_freq} = 20$, $\texttt{window\_size} = 10$, $\texttt{dimension} = 100$, $\texttt{maximum\_prototype} = 30$, $\alpha = 0.1$, $\texttt{epoch} = 1$, $\texttt{sense\_threshold} = 1e - 17$. It is worth noting that a large α (the underlying Dirichlet process) may lead to too many senses for some words and a small α, on the contrary, results in too few senses.

To measure the significance of the dialectal change, Kulkarni et al. (2015b) propose an unsupervised method to detect words with statistically significant meaning changes. However, given that we have access to the humanly curated `DialectSim` dataset, we evaluate the models on the list of annotated words using a simple thresh-holding model (where the thresh-hold parameter is learned from training data). Specifically, We evaluate both Skip-gram models (i.e. GEODIST and DIALECTGRAM) by calculating the Manhattan distance[6] between a word's region-specific embeddings[7].

5 Results

5.1 Qualitative Analysis

We investigate the words that GEODIST model predicts to have a significant dialectal change between the two regions. For example, the word *mate* is one of the top 20 words in our vocabulary if we sort the vocabulary by the Manhattan distance between the US and the UK embeddings from high to low. However, words like *draft* are predicted to have different regional meanings but not labelled as "significant" in `DialectSim`. We further discuss this issue in section 5.2.3.

We select some words with significantly different meanings between the UK and the US. In our DIALECTGRAM model, we select the most frequent 2 senses, which usually account for more than 99% usage variation of a word, and plot a heat map on world map.

The word maps in Figure [1, 2] suggest that the usage of *gas* and *flat* are different in the UK and in the US. *Gas* is used commonly as petrol and related to gas station in the US, but in the UK, *gas* usually refers to air and natural gas. *Flat* could refer to *apartment* but in the US this meaning is not as common as in the UK. The same model can also be used at a different resolution level (across US states). For example, given the word *buffalo*, we show the most dominant senses where *Buffalo City* (in blue) and the *buffalo sauce* sense (in white). Similarly for the word *pop*, we observe that the Midwest area and the Pacific Northwest are more reddish, indicating people are more likely to use the word for *soft drink, soda*, while people in other areas like to use it to describe a certain type of music – *pop music* [8].

5.2 Quantitative Results

Our training corpus `Geo-Tweets2019` has over three million tweets from US and UK. However, we still observed that micro-level analyses at a resolution lower than the state level required more data samples. Therefore, we only present the country-level and state-level analysis here (note that we do not need to train the model to learn embeddings again when we change resolutions for our analyses).

For each model, we defined a `score` function that takes in one word and return a real number denoting its difference in meanings between the UK and the US. We fit a simple threshold model that maximizes the accuracy on training set. Then we test the model performance on testing set. The results are shown in Table 4.

5.2.1 Frequency Model

We observed that Frequency Model is more sensitive to word difference between two countries: *football* in the UK is same as *soccer* in the US, causing an imbalanced frequency of term *football* between both countries. However, it can not detect some semantic changes of words if the semantic change preserves frequency for both countries: *flat* has similar frequency in both countries, despite the fact that *flat* could mean *apartment* in the UK, whereas this usage is uncommon in the US. This model does not suffer from an over-fitting problem, because the model is fairly simple and the parameter space is quite small. However the Frequency model is susceptible to a high false positive rate.

5.2.2 Syntactic Model

Syntactic Model performs the worst among all the models. It still gets slightly higher precision than

[6]We tried euclidean and cosine distance as well, but use Manhattan distance since it yielded the best results out of the three metrics.

[7]Our models, validation set and code are available at: https://github.com/yuxingch/DialectGram.

[8]We normalized the data points by filtering out states where the number of tweets is less than 15 since a small number of data points can suffer from high variance.

word	sense 1 neighbors	sense 2 neighbors
gas	industrial, masks, electric	car, station, bus
flat	kitchen, shower, window	shoes, problems, temperatures
buffalo	syracuse, hutchinson	chicken, fries, seafood
subway	starbucks, restaurant, mcdonalds	1mph, commercial, 5kmh

Table 3: Neighbors of sense embeddings for selected words. This shows DIALECTGRAM is able to learn semantic variations of words.

Model	Acc	Prec	Recall	F1
Frequency	0.5600	0.5600	0.5887	0.5568
Syntactic	0.5263	0.5714	0.4828	0.5233
GEODIST	0.6432	**0.7424**	0.5810	0.6518
DIALECTGRAM	**0.6667**	0.6837	**0.6438**	**0.6632**

Table 4: Test performance. Acc, Prec means accuracy and precision. DIALECTGRAM has better accuracy, recall, and F1 score than GEODIST.

the Frequency Model on test set because it gets some dialectal syntactic changes correct. There are two reasons for its bad performance. First, it is limited by the performance of POS Tagger. Second many word sense changes do not alter POS tags. For example, *pants* refers to *underwear* in the UK while it refers to *jeans* in the US, and both of them are nouns.

5.2.3 GEODIST Model

As mentioned in Section 5.1, GEODIST model is able to detect dialect changes. The accuracy on the test set beats the previous two baseline models (0.6432 versus 0.5600 and 0.5263), as shown in Table 4. It also outperforms the baseline models in terms of precision and F1 score. In fact, GEODIST model has the highest precision among all models, including the DIALECTGRAM model that will be discussed in the next section. We also notice that the recall on the test set is the lowest. The high precision with low recall indicates that for those changes that GEODIST model is very conservative and misses some words that actually have significant dialectal changes. For example, the difference between the two region-specific embeddings of the word *pants* is predicted to be not significant, while *pants* does have different meanings in the UK and the US (Table 1).

5.2.4 DialectGram Model

DialectGram outperforms the GEODIST model in accuracy, recall, and F1 score. However, its precision is lower than that of the GEODIST and Frequency Model. However, this is already impressive given the fact that DialectGram does not require pre-determined geographic labels and enables analysis at different geographic resolutions post-hoc (after the model is trained). One reason for DIALECTGRAM's lower performance in precision compared to GEODIST model is that it overestimates the number of senses (learning senses that overlap). For example the word *gas* in Table 3, we sometimes have an additional sense characterized by words such as *air, house, pipe*. This sense seems to be a mix of sense 1, gaseous substance, and sense 2, gasoline. The average number of senses is controlled by α which we pick based on the model's performance on the training set, but we acknowledge that smarter search strategies for α could be employed.

6 Conclusion

In this work, we proposed a novel method to detect linguistic variations on multiple resolution levels. In our new approach, we use DIALECTGRAM to train multiple sense embeddings on region-agnostic data, compose region-specific word embeddings, and determines whether there is a significant dialectal variation across regions for a word. In contrast to baseline models, DIALECTGRAM does not rely on the region-labels for training multi-sense word embeddings. The use of region-agnostic data allows DIALECTGRAM to conduct multi-resolution analysis with one-time training. We also construct Geo-Tweets2019, a new corpus from online Twitter users in the UK and US for training word embeddings. To validate our work, we also contribute a new validation set DialectSim for explicitly measuring the performance of our models in detecting the linguistic variations between the US and the UK. This validation set allows for more precise comparison between our method (DIALECTGRAM) and previous methods including Frequency Model, Syntactic Model, and GEODIST model. On DialectSim, our method achieves better performance than the previous models in accuracy, recall, and F1 score. Through linguistic analysis, we also found that DIALECTGRAM model learns rich linguistic changes between British and American English. Finally, we conclude by noting the method can be easily extended to temporal or analysis of language at multi-resolution levels.

Acknowledgments

We would like to thank Cindy Wang, Christopher Potts, and anonymous reviewers, who gave precious advice and comments to our paper. We would also like to thank Symbolic Systems Program at Stanford University for funding our research through Grants for Education And Research (GEAR).

References

David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138.

Léon Bottou. 1991. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*.

Gonzalo Donoso and David Sanchez. 2017. Dialectometric analysis of language variation in twitter. *arXiv preprint arXiv:1702.06777*.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287. Association for Computational Linguistics.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2014. Diffusion of lexical change in social media. *PloS one*, 9(11):e113114.

Jacob Eisenstein, Noah A Smith, and Eric P Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1365–1374. Association for Computational Linguistics.

Bruno Gonçalves and David Sánchez. 2016. Learning about spanish dialects through twitter. *Revista Internacional de Lingüística Iberoamericana*, pages 65–75.

Kristina Gulordava and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, pages 67–71.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*.

Adam Jatowt and Kevin Duh. 2014. A framework for analyzing semantic change of words across time. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 229–238. IEEE Press.

Tom Kenter, Melvin Wevers, Pim Huijnen, and Maarten De Rijke. 2015. Ad hoc monitoring of vocabulary shifts over time. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1191–1200. ACM.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515*.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015a. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee.

Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015b. Freshman or fresher? quantifying the geographic variation of internet language. *arXiv preprint arXiv:1510.06786*.

William Labov. 1980. *Locating language in time and space*. Academic Press New York.

Li Lucy and Julia Mendelsohn. 2018. Using sentiment induction to understand variation in gendered online communities. *arXiv preprint arXiv:1811.07061*.

James Milroy. 1992. *Linguistic variation and change: On the historical sociolinguistics of English*. B. Blackwell.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *HLT-NAACL*.

Philippa Shoemark, Farhana Ferdousi Liza, Dong Nguyen, Scott A Hale, and Barbara McGillivray. 2019. Room to glo: A systematic comparison of semantic change detection approaches with word embeddings.

Sali A Tagliamonte. 2006. *Analysing sociolinguistic variation*. Cambridge University Press.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *COLING*.

Walt Wolfram and Natalie Schilling. 2015. *American English: dialects and variation*, volume 25. John Wiley & Sons.

A principled derivation of Harmonic Grammar

Giorgio Magri
SFL, CNRS, University of Paris 8
magrigrg@gmail.com

Abstract

Phonologists focus on a few processes at the time. This practice is motivated by the intuition that phonological processes factorize into clusters with no interactions across clusters (e.g., obstruent voicing does not interact with vowel harmony). To formalize this intuition, we factorize a full-blown representation into under-specified representations, each encoding only the information needed by the corresponding phonological cluster. And we require a grammar for the original full-blown representations to factorize into grammars that handle the under-specified representations separately, independently of each other. Within a harmony-based implementation of constraint-based phonology, HG is shown to follow axiomatically from this grammar factorizability assumption.

1 Introduction

In constraint-based phonology, the best surface realization of an underlying form is the one with the smallest vector of constraint violations. How should constraint violation vectors be ordered to select the smallest? In other words, what is the proper model of constraint interaction? The literature has addressed this question by comparing competing ways of ordering constraint vectors on specific test cases. Yet, the predictions of a class of orderings on a specific test case depend on the choice of a specific constraint set. The conclusions reached are thus threatened to be overturned when a different constraint set is adopted.

An alternative, more principled approach starts instead from general formal properties that a grammar must satisfy in order to qualify as natural language phonology. And it deduces axiomatically from these desiderata what a suitable class of orderings of constraint violation vectors should look like. If this axiomatic deduction of the mode of constraint interaction holds independently of the constraint set, we will have untied the knot between the issue of determining the proper constraint set and the issue of characterizing the proper mode of constraint interaction.

This paper illustrates this research strategy. To set the background, section 2 recalls the framework of constraint-based phonology, independently of the choice of a specific mode of constraint interaction. Section 3 shows that a full-blown phonological representation often factorizes into multiple under-specified representations. And that these under-specified representations do not interact, in the sense that the constraint violations of the full-blown representations are simply the sum of the violations of the corresponding under-specified representations. In this case a grammar should factorize into multiple grammars that handle the under-specified representations separately, without these factor grammars interacting with each other. This factorizability condition formalizes the intuition that phonological processes factorize into small non-interacting clusters (e.g., obstruent voicing does not interact with vowel harmony), whereby phonologists can focus on a few processes at the time. Section 4 shows that a constraint-based grammar is indeed factorizable as long as the mode of constraint interaction satisfies a natural additivity condition. Section 5 finally shows that HG's weighted disharmony function can be derived axiomatically from this additivity condition. This result yields a principled justification of the HG mode of constraint interaction, which holds independently of any specific constraint set for any specific test case.

2 Constraint-based grammars

We assume that the core object of phonological theory is a **phonological mapping**, namely a pair (x, y) consisting of an underlying form x and a surface realization y (but see for instance Burzio 1996 for alternatives). To describe a specific phonological system, we start with a **representational framework** $\mathcal{R}$ that lists all the phonological mappings which are relevant for the system consid-

$$\mathcal{R} = \left\{ \begin{array}{cccc} (\text{/CV/, [CV]}) & (\text{/CV/, [CVC]}) & (\text{/CV/, [V]}) & (\text{/CV/, [VC]}) \\ (\text{/CVC/, [CV]}) & (\text{/CVC/, [CVC]}) & (\text{/CVC/, [V]}) & (\text{/CVC/, [VC]}) \\ (\text{/V/, [CV]}) & (\text{/V/, [CVC]}) & (\text{/V/, [V]}) & (\text{/V/, [VC]}) \\ (\text{/VC/, [CV]}) & (\text{/VC/, [CVC]}) & (\text{/VC/, [V]}) & (\text{/VC/, [VC]}) \end{array} \right\} \qquad \mathcal{C} = \left\{ \begin{array}{c} C_1 = \text{ONSET} \\ C_2 = \text{DEP} \\ C_3 = \text{CODA} \\ C_4 = \text{MAX} \end{array} \right\}$$

Figure 1: Representational framework $\mathcal{R}$ and constraint set $\mathcal{C}$ of the BSS (Prince and Smolensky 1993/2004).

ered. To illustrate, the representational framework $\mathcal{R}$ for the Basic Syllable System (BSS; Prince and Smolensky 1993/2004) consists of the sixteen mappings listed in figure 1.

We scan all the phonological mappings listed in a representational framework $\mathcal{R}$, extract their underlying forms, and collect them into the **base set** $B(\mathcal{R})$. To illustrate, the base of the BSS representational framework $\mathcal{R}$ in figure 1 consists of the four underlying syllable types /CV/, /CVC/, /V/, and /VC/. For every underlying form x in the base $B(\mathcal{R})$, we scan all the phonological mappings in $\mathcal{R}$ that feature this underlying form x and collect their surface forms into the **candidate set** $\mathcal{R}(\mathsf{x})$. To illustrate, the underlying forms of the BSS all share the candidate set consisting of the four surface syllable types [CV], [CVC], [V], and [VC].

A **constraint** C assigns to each phonological mapping (x, y) in the representational framework $\mathcal{R}$ a number $C(\mathsf{x}, \mathsf{y})$. This number $C(\mathsf{x}, \mathsf{y})$ is interpreted as a count of some undesirable phonological structure: an offending cluster, a mismatch between corresponding segments, etcetera. This number $C(\mathsf{x}, \mathsf{y})$ is thus assumed to be a nonnegative integer. This **constraint integrality assumption** captures the intuition that the properties relevant for phonology (contrary to phonetics) are discrete in nature. This assumption will play a crucial role in Section 5. We assume a set $\mathcal{C}$ consisting of a finite number n of constraints $C_1, \ldots, C_n$. It effectively represents a mapping (x, y) as the n-dimensional **constraint violation vector** $\mathbf{C}(\mathsf{x}, \mathsf{y}) = (C_1(\mathsf{x}, \mathsf{y}), \ldots, C_n(\mathsf{x}, \mathsf{y}))$. We denote by $\mathcal{C}(\mathcal{R})$ the set of the constraint vectors of all mappings in the representational framework $\mathcal{R}$. To illustrate, a constraint set for the BSS consists of the $n = 4$ constraints listed in figure 1.

The underlying and surface forms in the representational framework $\mathcal{R}$ are **discrete** objects (but see Smolensky, Goldrick, and Mathis 2014): finite strings constructed out of a finite number of discrete segments, or auto-segmental graphs constructed out of a finite number of feature values, etcetera. Dealing with discrete objects is difficult because only very little "structure" is defined on them. To circumvent this difficulty, constraint-based phonology "represents" the discrete phonological mappings in $\mathcal{R}$ as the set $\mathcal{C}(\mathcal{R})$ of numerical constraint violation vectors and thus imports into phonology the rich structure defined on numbers and vectors thereof (Haussler 1999).

For instance, numbers can be ordered based on their size. This ordering can be extended from single numbers to vectors in many different ways. Thus, let $\prec$ be some order defined among n-dimensional vectors. The inequality $\mathbf{a} \prec \mathbf{b}$ says that the vector **a** is **smaller** than the vector **b**. The **constraint-based grammar** (CBG) corresponding to this order $\prec$ is the function $G_\prec = G_\prec^{\mathcal{R},\mathcal{C}}$ that realizes each underlying form x in the base $B(\mathcal{R})$ as the surface form $\mathsf{y} = G_\prec(\mathsf{x})$ in the candidate set $\mathcal{R}(\mathsf{x})$ with the smallest constraint violation vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$. That is, the inequality $\mathbf{C}(\mathsf{x}, \mathsf{y}) \prec \mathbf{C}(\mathsf{x}, \mathsf{z})$ holds for every other candidate z in $\mathcal{R}(\mathsf{x})$ (we assume that such a candidate y always exists).

To illustrate, let us consider an arbitrary subset $S \subseteq \{1, ..., n\}$ that singles out the dimensions/constraints that are deemed relevant. The relation $\prec_S$ defined in (1) for any two vectors $\mathbf{a} = (a_1, ..., a_n)$ and $\mathbf{b} = (b_1, ..., b_n)$ is a partial order among n-dimensional vectors.

$$\mathbf{a} \prec_S \mathbf{b} \text{ iff } a_k \leq b_k \text{ for every } k \in S \tag{1}$$

We focus on the representational framework $\mathcal{R}$ and the constraint set $\mathcal{C}$ for the BSS in figure 1. We focus next on the order $\prec_S$ among 4-dimensional vectors corresponding to the set $S = \{C_1, C_3\}$. The corresponding CBG $G_{\prec_S}^{\mathcal{R},\mathcal{C}}$ maps all underlying forms to [CV]. This makes sense: if only the two markedness constraints C_1 = ONSET and C_3 = CODA are singled out as relevant by the set S, the smallest constraint vector is always the one corresponding to the unmarked surface form [CV].

3 Factorizable representations

3.1 Underspecification

A phonological representation x encodes a certain amount of phonological information. Often, this information can be split into two representations x′

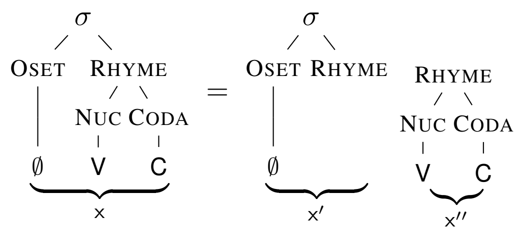
$$
\mathcal{R} = \left\{ \begin{array}{cccc}
(/\mathsf{CV}/, [\mathsf{CV}]) & (/\mathsf{CV}/, [\mathsf{V}]) & (/\mathsf{V}/, [\mathsf{V}]) & (/\mathsf{V}/, [\mathsf{CV}]) \\
(/\mathsf{CV}/, [\mathsf{CVC}]) & (/\mathsf{CV}/, [\mathsf{VC}]) & (/\mathsf{V}/, [\mathsf{VC}]) & (/\mathsf{V}/, [\mathsf{CVC}]) \\
(/\mathsf{CVC}/, [\mathsf{CVC}]) & (/\mathsf{CVC}/, [\mathsf{VC}]) & (/\mathsf{VC}/, [\mathsf{VC}]) & (/\mathsf{VC}/, [\mathsf{CVC}]) \\
(/\mathsf{CVC}/, [\mathsf{CV}]) & (/\mathsf{CVC}/, [\mathsf{V}]) & (/\mathsf{VC}/, [\mathsf{V}]) & (/\mathsf{VC}/, [\mathsf{CV}])
\end{array} \right\}
\begin{array}{c} \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \end{array}
\left\{ \begin{array}{c}
(/\square\mathsf{V}/, [\square\mathsf{V}]) \\
(/\square\mathsf{V}/, [\square\mathsf{VC}]) \\
(/\square\mathsf{VC}/, [\square\mathsf{VC}]) \\
(/\square\mathsf{VC}/, [\square\mathsf{V}])
\end{array} \right\} = \mathcal{R}''
$$

$$
\mathcal{R}' = \left\{ \begin{array}{cccc}
(/\mathsf{CV}\square/, [\mathsf{CV}\square]) & (/\mathsf{CV}\square/, [\mathsf{V}\square]) & (/\mathsf{V}\square/, [\mathsf{V}\square]) & (/\mathsf{V}\square/, [\mathsf{CV}\square])
\end{array} \right\}
$$

Figure 2: Factorization of the representational framework $\mathcal{R}$ of the BSS into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ under-specified for codas and for onsets, respectively

and x''. These two representations x' and x'' *individually* encode less information than the original representation x. In other words, they are **under-specified** relative to the original representation (Steriade 1995). Yet, these two under-specified representations x' and x'' *together* encode the same information as the full-blown representation x. In other words, the full-blown representation x **factorizes** into these two under-specified representations x' and x'', whereby we write $\mathsf{x} = \mathsf{x}'\mathsf{x}''$.

To illustrate again with the BSS, we note that a syllable type such as VC can be represented as the tree x on the left hand side of (2). This tree comes with the two sub-trees x' and x'' on the right hand side. These sub-trees can be interpreted as representations underspecified for codas and for onsets, respectively. We will denote these sub-trees compactly as $\mathsf{V}\square$ and $\square\mathsf{VC}$. The full-blown syllable $\mathsf{x} = \mathsf{VC}$ thus factorizes into these two under-specified representations $\mathsf{x}' = \mathsf{V}\square$ and $\mathsf{x}'' = \square\mathsf{VC}$.



(2)

Feature-based phonology provides a natural strategy to factorize full-blown representations into under-specified representations. For instance, the round mid vowel can be represented as the tuple of feature values $\mathsf{x} = [+\mathsf{round} - \mathsf{high} - \mathsf{low}]$. This tuple comes with sub-tuples such as $\mathsf{x}' = [+\mathsf{round}]$ and $\mathsf{x}'' = [-\mathsf{high} - \mathsf{low}]$. These sub-tuples can be interpreted as representations under-specified for height and for rounding, respectively. The full-blown vowel x thus factorizes into these two under-specified representations x' and x''.

$$\underbrace{[+\mathsf{round}\ -\mathsf{high}\ -\mathsf{low}]}_{\mathsf{x}} = \underbrace{[+\mathsf{round}]}_{\mathsf{x}'}\underbrace{[-\mathsf{high}\ -\mathsf{low}]}_{\mathsf{x}''} \quad (3)$$

3.2 Representational assumptions

A framework $\mathcal{R}$ of full-blown representations **factorizes** into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations provided $\mathcal{R}$ is the set of all and only the mappings $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$ that factorize into two mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$ from $\mathcal{R}'$ and $\mathcal{R}''$, as in (4). In the sense that $\mathsf{x}'\mathsf{x}''$ and $\mathsf{y}'\mathsf{y}''$ are underlying and surface full-blown representations that factorize into the underlying and surface under-specified representations $\mathsf{x}', \mathsf{x}''$ and $\mathsf{y}', \mathsf{y}''$.

$$\mathcal{R} = \mathcal{R}'\mathcal{R}'' = \left\{ (\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'') \,\middle|\, \begin{array}{l} (\mathsf{x}', \mathsf{y}') \in \mathcal{R}' \\ (\mathsf{x}'', \mathsf{y}'') \in \mathcal{R}'' \end{array} \right\} \quad (4)$$

Equivalently, the base of the full-blown representational framework $\mathcal{R}$ factorizes into the bases of the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$, namely $B(\mathcal{R}) = B(\mathcal{R}')B(\mathcal{R}'')$. And the candidate sets of $\mathcal{R}$ factorize into the corresponding candidate sets of $\mathcal{R}'$ and $\mathcal{R}''$, namely $\mathcal{R}(\mathsf{x}'\mathsf{x}'') = \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$.

To illustrate, let us consider again the representational framework $\mathcal{R}$ for the BSS in figure 1. We consider next the representational framework $\mathcal{R}'$ that consists of the four mappings that can be assembled out of the two representations $\mathsf{CV}\square$ and $\mathsf{V}\square$ that specify whether the onset is filled or empty but are under-specified for codas. And the representational framework $\mathcal{R}''$ that consists of the four mappings that can be assembled out of the two representations $\square\mathsf{V}$ and $\square\mathsf{VC}$ that specify whether the coda is filled or empty but are under-specified for onsets. As indicated by the dotted lines in figure 2, each full-blown mapping in $\mathcal{R}$ factorizes into the two under-specified mappings in $\mathcal{R}'$ and $\mathcal{R}''$ that sit in the same column and the same row. We conclude that condition (4) holds and that the framework $\mathcal{R}$ of full-blown syllable representations therefore factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of syllable representations under-specified for codas and for onsets.

As a second example, let us consider the representational framework $\mathcal{R}$ consisting of the 36

$$
\mathcal{R} = \left\{ \begin{array}{cccc}
(/\mathrm{i}/, [\mathrm{i}]) & (/\mathrm{i}/, [\mathrm{u}]) & (/\mathrm{u}/, [\mathrm{i}]) & (/\mathrm{u}/, [\mathrm{u}]) \\
(/\mathrm{i}/, [\mathrm{e}]) & (/\mathrm{i}/, [\mathrm{ɒ}]) & (//\mathrm{u}, [\mathrm{a}]) & (/\mathrm{u}/, [\mathrm{ɒ}]) \\
(/\mathrm{i}/, [\mathrm{a}]) & (/\mathrm{i}/, [\mathrm{ɒ}]) & (/\mathrm{u}/, [\mathrm{a}]) & (/\mathrm{u}/, [\mathrm{ɒ}]) \\
(/\mathrm{e}/, [\mathrm{i}]) & (/\mathrm{e}/, [\mathrm{u}]) & (/\mathrm{o}/, [\mathrm{i}]) & (/\mathrm{o}/, [\mathrm{u}]) \\
(/\mathrm{e}/, [\mathrm{e}]) & (/\mathrm{e}/, [\mathrm{o}]) & (/\mathrm{o}/, [\mathrm{e}]) & (/\mathrm{o}/, [\mathrm{o}]) \\
(/\mathrm{e}/, [\mathrm{a}]) & (/\mathrm{e}/, [\mathrm{ɒ}]) & (/\mathrm{o}/, [\mathrm{a}]) & (/\mathrm{o}/, [\mathrm{ɒ}]) \\
(/\mathrm{a}/, [\mathrm{i}]) & (/\mathrm{a}/, [\mathrm{u}]) & (/\mathrm{ɒ}/, [\mathrm{i}]) & (/\mathrm{ɒ}/, [\mathrm{u}]) \\
(/\mathrm{a}/, [\mathrm{e}]) & (/\mathrm{a}/, [\mathrm{o}]) & (/\mathrm{ɒ}/, [\mathrm{e}]) & (/\mathrm{ɒ}/, [\mathrm{o}]) \\
(/\mathrm{a}/, [\mathrm{a}]) & (/\mathrm{a}/, [\mathrm{ɒ}]) & (/\mathrm{ɒ}/, [\mathrm{a}]) & (/\mathrm{ɒ}/, [\mathrm{ɒ}]) \\
\vdots & \vdots & \vdots & \vdots
\end{array} \right\}
\begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{array}
\left\{ \begin{array}{c}
(/{+}\mathrm{high},\ {-}\mathrm{low}/, [{+}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{+}\mathrm{high},\ {-}\mathrm{low}/, [{-}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{+}\mathrm{high},\ {-}\mathrm{low}/, [{-}\mathrm{high},\ {+}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {-}\mathrm{low}/, [{+}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {-}\mathrm{low}/, [{-}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {-}\mathrm{low}/, [{-}\mathrm{high},\ {+}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {+}\mathrm{low}/, [{+}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {+}\mathrm{low}/, [{-}\mathrm{high},\ {-}\mathrm{low}]) \\
(/{-}\mathrm{high},\ {+}\mathrm{low}/, [{-}\mathrm{high},\ {+}\mathrm{low}])
\end{array} \right\} = \mathcal{R}''
$$

$$
\mathcal{R}' = \Big\{ (/{-}\mathrm{rnd}/, [{-}\mathrm{rnd}]) \quad (/{-}\mathrm{rnd}/, [{+}\mathrm{rnd}]) \quad (/{+}\mathrm{rnd}/, [{-}\mathrm{rnd}]) \quad (/{+}\mathrm{rnd}/, [{+}\mathrm{rnd}]) \Big\}
$$

Figure 3: Factorization of the representational framework $\mathcal{R}$ for full-blown vowels into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ under-specified for height and for rounding, respectively

mappings that can be assembled out of the six vowels a, e, i, ɒ, o, and u. We consider next the representational framework $\mathcal{R}'$ that consists of the four mappings that can be assembled out of the two representations [+round] and [−round] underspecified for height. And the representational framework $\mathcal{R}''$ that consists of the nine mappings that can be assembled out of the the three representations [+high, −low], [−high, −low], and [−high, +low] underspecified for rounding. As indicated by the dotted lines in figure 3, each full-blown mapping in $\mathcal{R}$ factorizes into the two under-specified mappings in $\mathcal{R}'$ and $\mathcal{R}''$ that sit in the same column and the same row. We conclude that condition (4) holds and that the framework $\mathcal{R}$ of full-blown vowel representations therefore factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of vowel representations under-specified for height and for rounding.

3.3 No interaction

We consider a constraint set $\mathcal{C}$ for the mappings (x, y) in the full-blown representational framework $\mathcal{R}$. We assume that each constraint C in this constraint set $\mathcal{C}$ can be extended to the mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$ in the under-specified factor representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that condition (5) holds. It says that the number of violations $C(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y})$ that a constraint C assigns to a full-blown mapping $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y})$ is the sum of the number of violations $C(\mathsf{x}', \mathsf{y}')$ and $C(\mathsf{x}'', \mathsf{y}'')$ that it assigns to the two under-specified factor mappings $(\mathsf{x}', \mathsf{y}')$ and $(\mathsf{x}'', \mathsf{y}'')$. In other words, no violations are created nor lost when the under-specified representations are assembled together into the full-blown representations.

$$
C(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}) = C(\mathsf{x}', \mathsf{y}') + C(\mathsf{x}'', \mathsf{y}'') \qquad (5)
$$

Suppose that this condition (5) holds for every underlying representation $\mathsf{x}'\mathsf{x}''$ in the base set $B(\mathcal{R}) = B(\mathcal{R}')B(\mathcal{R}'')$, for every surface representation $\mathsf{y}'\mathsf{y}''$ in the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'') = \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$, and for every constraint C in the constraint set $\mathcal{C}$. In other words, the set $\mathcal{C}(\mathcal{R})$ of constraint vectors of $\mathcal{R}$ is the sum of the sets $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ of constraint vectors of $\mathcal{R}'$ and $\mathcal{R}''$, namely $\mathcal{C}(\mathcal{R}) = \mathcal{C}(\mathcal{R}') + (\mathcal{R}'')$. In this case, we say that the two under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ **do not interact** relative to the constraint set $\mathcal{C}$.

To illustrate, let us consider again the representational framework $\mathcal{R}$ and the constraint set $\mathcal{C}$ for the BSS in figure 1. The set $\mathcal{C}(\mathcal{R})$ of the constraint violation vectors of the sixteen mappings in the representational framework $\mathcal{R}$ is listed in figure 4. We extend the $n = 4$ constraints to the under-specified mappings in the two factor representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ straightforwardly as follows. The constraint C_1 = ONSET assigns zero violations to the four mappings in the factor representational framework $\mathcal{R}''$ under-specified for onsets. The constraint C_3 = CODA assigns zero violations to the four mappings in the other factor representational framework $\mathcal{R}'$ under-specified for codas. The other two constraints C_2 = DEP and C_4 = MAX simply count epenthetic and deleted consonants and thus assign violations to mappings in both factor representational frameworks. The corresponding sets $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ of constraint vectors are listed at the bottom and on the left of figure 4. As indicated by the dotted lines, each constraint violation vector in $\mathcal{C}(\mathcal{R})$ is the (component-wise) sum of the constraint violation vectors in $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$ that sit in the

$$
\mathcal{C}(\mathcal{R}) = \left\{
\begin{array}{ccccc}
 & \text{(/CV/,[CV])} & \text{(/CV/, [CVC])} & \text{(/CVC/, [CV])} & \text{(/CVC/, [CVC])} \\
\begin{array}{r}\text{Onset}\\ \text{Dep}\\ \text{Coda}\\ \text{Max}\end{array} & \begin{bmatrix}0\\0\\0\\0\end{bmatrix} & \begin{bmatrix}0\\1\\1\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\end{bmatrix} & \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \\
 & \text{(/CV/, [V])} & \text{(/CV/, [VC])} & \text{(/CVC/, [V])} & \text{(/CVC/, [VC])} \\
 & \begin{bmatrix}1\\0\\0\\1\end{bmatrix} & \begin{bmatrix}1\\1\\1\\1\end{bmatrix} & \begin{bmatrix}1\\0\\0\\2\end{bmatrix} & \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \\
 & \text{(/V/, [V])} & \text{(/V/, [VC])} & \text{(/VC/, [V])} & \text{(/VC/, [VC])} \\
 & \begin{bmatrix}1\\0\\0\\0\end{bmatrix} & \begin{bmatrix}1\\1\\1\\0\end{bmatrix} & \begin{bmatrix}1\\0\\0\\1\end{bmatrix} & \begin{bmatrix}1\\0\\1\\0\end{bmatrix} \\
 & \text{(/V/, [CV])} & \text{(/V/, [CVC])} & \text{(/VC/, [CV])} & \text{(/VC/, [CVC])} \\
 & \begin{bmatrix}0\\1\\0\\0\end{bmatrix} & \begin{bmatrix}0\\2\\1\\0\end{bmatrix} & \begin{bmatrix}0\\1\\0\\1\end{bmatrix} & \begin{bmatrix}0\\1\\1\\0\end{bmatrix}
\end{array}
\right\}
\cdots\cdots
\left\{
\begin{array}{c}
\text{(/CV}\square\text{/, [CV}\square\text{])} \\
\begin{bmatrix}0\\0\\0\\0\end{bmatrix} \\
\text{(/CV}\square\text{/, [V}\square\text{])} \\
\begin{bmatrix}1\\0\\0\\1\end{bmatrix} \\
\text{(/V}\square\text{/, [V}\square\text{])} \\
\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \\
\text{(/V}\square\text{/, [CV}\square\text{])} \\
\begin{bmatrix}0\\1\\0\\0\end{bmatrix}
\end{array}
\right\} = \mathcal{C}(\mathcal{R}'')
$$

$$
\vdots \qquad \vdots \qquad \vdots \qquad \vdots
$$

$$
\mathcal{C}(\mathcal{R}') = \left\{
\begin{array}{cccc}
\text{(/}\square\text{V/, [}\square\text{V])} & \text{(/}\square\text{V/, [}\square\text{VC])} & \text{(/}\square\text{VC/, [}\square\text{V])} & \text{(/}\square\text{VC/, [}\square\text{VC])} \\
\begin{bmatrix}0\\0\\0\\0\end{bmatrix} & \begin{bmatrix}0\\1\\1\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\end{bmatrix} & \begin{bmatrix}0\\0\\1\\0\end{bmatrix}
\end{array}
\right\}
$$

Figure 4: The constraints for the representational framework $\mathcal{R}$ of the BSS can be extended to the factor frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the constraint vectors in $\mathcal{C}(\mathcal{R})$ are the sums of the constraint vectors in $\mathcal{C}(\mathcal{R}')$ and $\mathcal{C}(\mathcal{R}'')$.

same column and the same row. We conclude that condition (5) holds and that the two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ therefore do not interact relative to the constraint set $\mathcal{C}$.

4 Factorizable grammars

4.1 Factorizability

We consider a CBG $G_\prec = G_\prec^{\mathcal{R},\mathcal{C}}$ corresponding to some representational framework $\mathcal{R}$, some set $\mathcal{C}$ of n constraints for this representational framework $\mathcal{R}$, and some order $\prec$ among n-dimensional vectors. We assume that the full-blown representational framework $\mathcal{R}$ factorizes into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations and we consider some suitable extension of the constraint set $\mathcal{C}$ to $\mathcal{R}'$ and $\mathcal{R}''$. Using the same vector order $\prec$, we construct the CBGs $G'_\prec = G_\prec^{\mathcal{R}',\mathcal{C}}$ and $G''_\prec = G_\prec^{\mathcal{R}'',\mathcal{C}}$ for the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$. We say that the original grammar $G_\prec$ **factorizes** into the two grammars $G'_\prec$ and $G''_\prec$ provided the identity (6) holds for any under-specified underlying representations x' and x'' in the base sets $B(\mathcal{R}')$ and $B(\mathcal{R}'')$ (see also Magri 2013). In this case, we also write $G_\prec = G'_\prec G''_\prec$.

$$
G_\prec(\mathsf{x}'\mathsf{x}'') = G'_\prec(\mathsf{x}')G''_\prec(\mathsf{x}'') \tag{6}
$$

This identity (6) says that an underlying representation $\mathsf{x}'\mathsf{x}''$ that factorizes into two under-specified underlying representations x' and x'' admits a surface realization $G_\prec(\mathsf{x}'\mathsf{x}'')$ that itself factorizes into the two under-specified surface representations $G'_\prec(\mathsf{x}')$ and $G''_\prec(\mathsf{x}'')$. In other words, the job done by the grammar $G_\prec$ can be outsourced to two grammars $G'_\prec$ and $G''_\prec$ that each carry out half of it independently from the other.

To illustrate, we consider again the representational framework $\mathcal{R}$ for the BSS and its factors $\mathcal{R}'$ and $\mathcal{R}''$ in figure 2. The grammar G in figure 5 tolerates empty onsets but deletes codas. We consider next the grammar G' for representations under-specified for codas that tolerates empty onsets. And the grammar G'' for representations under-specified for onsets that deletes codas. As indicated by the dotted lines, each full-blown mapping in G factorizes into the two under-specified mappings in G' and G'' that sit in the same column

$$G=\left\{\begin{array}{ll}(\text{/CV/, [CV]}) & (\text{/V/, [V]}) \\ (\text{/CVC/, [CV]}) & (\text{/VC/, [V]}) \\ \vdots & \vdots \end{array}\right\}\cdots\left\{\begin{array}{l}(\text{/□V/, [□V]}) \\ (\text{/□VC/, [□V]})\end{array}\right\}=G''$$

$$G'=\left\{(\text{/CV□/, [CV□]})\ (\text{/V□/, [V□]})\right\}$$

Figure 5: Factorization of the grammar G into two grammars G' and G''.

and the same row. We conclude that condition (6) holds and that the grammar G for full-blown syllable representations therefore factorizes into the two grammars G' and G'' for syllable representations under-specified for codas and for onsets.

Consider instead the grammar G in figure 6. It tolerates empty onsets and codas as long as they do not co-occur, as /VC/ is neutralized to [V] rather than faithfully realized as [VC]. This grammar does not factorize: onsets and codas cannot be handled independently. Indeed, it is easy to verify that, no matter what we replace the red question mark in figure 6 with, the factorizability identity (6) fails. This grammar G in figure 6 would be easy to get as a CBG corresponding to a markedness constraint set that contains a constraint that selectively penalizes the doubly-marked syllable type [VC]. But such a constraint does not satisfy the constraint condition (5): it would not penalize the underspecified surface representations y$'$ = [V□] and y$''$ = [□VC] but it would penalize the corresponding full-blown representation y$'$y$''$ = [VC]. In other words, the two underspecified representations do interact relative to such a constraint set.

4.2 Additive orders

An order $\prec$ among n-dimensional vectors is **additive** provided it satisfies the implication (7) for any three vectors **a**, **b**, and **c** (Anderson and Feil 1988). This implication (7) captures the intuition that, if **a** is smaller than **b** and if the same quantity **c** is added to both, the resulting sum $\mathbf{a}+\mathbf{c}$ ought to be smaller than the sum $\mathbf{b}+\mathbf{c}$ (all vector sums are component-wise).

$$\text{If } \mathbf{a} \prec \mathbf{b}, \text{ then } \mathbf{a}+\mathbf{c} \prec \mathbf{b}+\mathbf{c}. \qquad (7)$$

To illustrate, this additivity condition (7) is satisfied by the vector order $\prec_S$ defined in (1), for any choice of the set $S \subseteq \{1,\ldots,n\}$. Although this additivity condition (7) feels intuitive, it easy to construct orders that flout it. To illustrate, let $\mathbf{a} \prec \mathbf{b}$ provided the sum of squared components of the vector $\mathbf{a} = (a_1,\ldots,a_n)$ is smaller than the sum of squared components of the vector $\mathbf{b} = (b_1,\ldots,b_n)$, namely $a_1^2+\ldots+a_n^2 < b_1^2+\ldots+b_n^2$. The resulting order $\prec$ is not additive.

$$G=\left\{\begin{array}{ll}(\text{/CV/, [CV]}) & (\text{/V/, [V]}) \\ (\text{/CVC/, [CVC]}) & (\text{/VC/, [V]}) \\ \vdots & \vdots \end{array}\right\}\cdots\left\{\begin{array}{l}(\text{/□V/, [□V]}) \\ (\text{/□VC/, ??})\end{array}\right\}=G''$$

$$G'=\left\{(\text{/CV□/, [CV□]})\ (\text{/V□/, [V□]})\right\}$$

Figure 6: An example of grammar G that does not factorize into two grammars G' and G''.

4.3 Establishing factorizability

The following proposition says that additivity of a vector order is sufficient to ensure that the corresponding CBG factorizes (see Prince 2015 for a special case of this result; see Magri and Storme 2020 for a different phonological justification of additive vector orders). Additivity can also be shown to be necessary, in the sense that for any order which is not additive we can construct a corresponding CBG that fails to factorize. Additivity thus provides a complete answer to the problem of characterizing grammatical factorizability.

Proposition 1 *Consider a framework $\mathcal{R}$ of full-blown representations that factorizes into two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of under-specified representations, namely $\mathcal{R} = \mathcal{R}'\mathcal{R}''$ in the sense of condition (4) in subsection 3.2. Consider a set $\mathcal{C}$ of n constraints for the full-blown framework $\mathcal{R}$ that can be extended to the two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the additivity condition (5) in subsection 3.3 holds. Finally, consider an order $\prec$ among n-dimensional vectors that satisfies the additivity condition (7) in subsection 4.2. The corresponding CBG $G_{\prec}^{\mathcal{R},\mathcal{C}}$ for the full-blown representational framework $\mathcal{R}$ then factorizes into the two CBGs $G_{\prec}^{\mathcal{R}',\mathcal{C}}$ and $G_{\prec}^{\mathcal{R}'',\mathcal{C}}$ for the under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$.* □

To illustrate, we have seen in figure 2 that the representational framework $\mathcal{R}$ for the BSS factorizes into the two frameworks $\mathcal{R}'$ and $\mathcal{R}''$ of syllable representations under-specified for codas and for onsets, respectively. Furthermore, we have seen in figure 4 that the constraint set $\mathcal{C}$ for the BSS can be extended to these two under-specified frameworks $\mathcal{R}'$ and $\mathcal{R}''$ in such a way that the additivity condition (5) holds. Finally, we have seen in subsection 4.2 that the vector order $\prec_S$ defined in (1) is additive for any subset S. Proposition 1 thus ensures that the CBG $G^{\mathcal{R},\mathcal{C}}_{\prec_S}$ factorizes.

4.4 Proof of proposition 1

Let us suppose that the two CBGs $G'_{\prec} = G^{\mathcal{R}',\mathcal{C}}_{\prec}$ and $G''_{\prec} = G^{\mathcal{R}'',\mathcal{C}}_{\prec}$ realize the under-specified underlying strings x' and x'' as the under-specified surface strings y' and y'' in the candidate sets $\mathcal{R}'(\mathsf{x}')$ and $\mathcal{R}''(\mathsf{x}'')$, namely $G'_{\prec}(\mathsf{x}') = \mathsf{y}'$ and $G''_{\prec}(\mathsf{x}'') = \mathsf{y}''$. The full-blown surface representation $\mathsf{y}'\mathsf{y}''$ belongs to the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'')$ because of the inclusion $\mathcal{R}(\mathsf{x}'\mathsf{x}'') \supseteq \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$. We need to show that $\mathsf{y}'\mathsf{y}''$ is indeed the surface realization of the full-blown underlying representation $\mathsf{x}'\mathsf{x}''$ according to the CBG $G_{\prec} = G^{\mathcal{R},\mathcal{C}}_{\prec}$, namely $G_{\prec}(\mathsf{x}'\mathsf{x}'') = \mathsf{y}'\mathsf{y}''$.

To this end, let us consider a candidate z in the candidate set $\mathcal{R}(\mathsf{x}'\mathsf{x}'')$ different from the candidate $\mathsf{y}'\mathsf{y}''$. This candidate z must factorize as $\mathsf{z} = \mathsf{z}'\mathsf{z}''$ into some candidates z' and z'' from $\mathcal{R}'(\mathsf{x}')$ and $\mathcal{R}''(\mathsf{x}'')$, because of the inclusion $\mathcal{R}(\mathsf{x}'\mathsf{x}'') \subseteq \mathcal{R}'(\mathsf{x}')\mathcal{R}''(\mathsf{x}'')$. The assumption $\mathsf{z} \neq \mathsf{y}'\mathsf{y}''$ means that $\mathsf{z}' \neq \mathsf{y}'$ or $\mathsf{z}'' \neq \mathsf{y}''$ (or both). Without loss of generality, we assume $\mathsf{z}' \neq \mathsf{y}'$.

Since $\mathsf{z}' \neq \mathsf{y}'$, the assumption $G'_{\prec}(\mathsf{x}') = \mathsf{y}'$ says that the constraint violation vector $\mathbf{C}(\mathsf{x}', \mathsf{y}')$ of the winner mapping $(\mathsf{x}', \mathsf{y}')$ is smaller than the constraint violation vector $\mathbf{C}(\mathsf{x}', \mathsf{z}')$ of the loser mapping $(\mathsf{x}', \mathsf{z}')$, as in (8).

$$\mathbf{C}(\mathsf{x}', \mathsf{y}') \prec \mathbf{C}(\mathsf{x}', \mathsf{z}') \tag{8}$$

Let us now turn to the other two candidates y'' and z''. If they are different as well, we reason analogously that the constraint violation vector $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ of the winner mapping $(\mathsf{x}'', \mathsf{y}'')$ must be smaller than the constraint violation vector $\mathbf{C}(\mathsf{x}'', \mathsf{z}'')$ of the loser mapping $(\mathsf{x}'', \mathsf{z}'')$, as in (9).

$$\mathbf{C}(\mathsf{x}'', \mathsf{y}'') \prec \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \tag{9}$$

If instead these two candidates y'' and z'' are identical, their constraint violation vectors $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ and $\mathbf{C}(\mathsf{x}'', \mathsf{z}'')$ coincide, as stated in (10).

$$\mathbf{C}(\mathsf{x}'', \mathsf{y}'') = \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \tag{10}$$

Since the order $\prec$ satisfies the additivity condition (7), the inequality (8) and the identity (10) can be summed together into the inequality (11).

$$\mathbf{C}(\mathsf{x}', \mathsf{y}') + \mathbf{C}(\mathsf{x}'', \mathsf{y}'') \prec \mathbf{C}(\mathsf{x}', \mathsf{z}') + \mathbf{C}(\mathsf{x}'', \mathsf{z}'') \tag{11}$$

Suppose instead that it is the inequality (9) that holds rather than the identity (10). In this case, we note that the additivity condition (7) entails the variant in (12) for any four vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$. In fact, the assumption $\mathbf{a} \prec \mathbf{b}$ in the antecedent of (12) ensures that $\mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{c}$ through the additivity condition (7). Analogously, the assumption $\mathbf{c} \prec \mathbf{d}$ ensures that $\mathbf{b} + \mathbf{c} \prec \mathbf{b} + \mathbf{d}$. The consequent $\mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{d}$ then follows by transitivity of $\prec$.

$$\text{If } \mathbf{a} \prec \mathbf{b} \text{ and } \mathbf{c} \prec \mathbf{d}, \text{ then } \mathbf{a} + \mathbf{c} \prec \mathbf{b} + \mathbf{d} \tag{12}$$

Since the vector order $\prec$ satisfies condition (12), the inequalities (8) and (9) can be summed together yielding once again the inequality (11).

By assumption, the two under-specified representational frameworks $\mathcal{R}'$ and $\mathcal{R}''$ do not interact relative to the constraint set $\mathcal{C}$, in the sense of condition (5). Thus, the sum of the constraint violation vectors $\mathbf{C}(\mathsf{x}', \mathsf{y}')$ and $\mathbf{C}(\mathsf{x}'', \mathsf{y}'')$ on the left hand side of the inequality (11) coincides with the constraint violation vector $\mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$ of the corresponding full-blown mapping $(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'')$. Analogously for the right hand side, whereby the inequality (11) can be rewritten as (13).

$$\mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{y}'\mathsf{y}'') \prec \mathbf{C}(\mathsf{x}'\mathsf{x}'', \mathsf{z}'\mathsf{z}'') \tag{13}$$

By (13), the constraint violation vector of the candidate $\mathsf{y}'\mathsf{y}''$ is smaller than that of any competing candidate $\mathsf{z} = \mathsf{z}'\mathsf{z}''$. The CBG $G_{\prec}$ thus realizes the full-blown underlying representation $\mathsf{x}'\mathsf{x}''$ as the full-blown surface representation $\mathsf{y}'\mathsf{y}''$, namely $G_{\prec}(\mathsf{x}'\mathsf{x}'') = \mathsf{y}'\mathsf{y}''$ as desired.

5 HG and factorizability

5.1 Disharmony functions

Let us consider a particularly natural way of ordering numerical vectors. We start from a function H that assigns to each vector $\mathbf{a}$ a number $H(\mathbf{a})$ called its **disharmony**. Any two vectors $\mathbf{a}$ and $\mathbf{b}$ can then be ordered based on their disharmonies $H(\mathbf{a})$ and $H(\mathbf{b})$ as in (14): the smaller (and thus better) vector is the one with the smaller disharmony.

$$\mathbf{a} \prec_H \mathbf{b} \text{ iff } H(\mathbf{a}) < H(\mathbf{b}) \tag{14}$$

The disharmony function H thus effectively defines a partial strict order $\prec_H$ among vectors.

Crucially, there exist numerical orders that are not induced by any disharmony function H. In the sense that condition (14) fails for some vectors, no matter how the disharmony function H is chosen. For instance, that can be shown to be case for the vector order $\prec_S$ defined in (1), whenever the set S has cardinality larger than one. The restriction to vector orders that are induced by disharmony functions is therefore substantive.

5.2 Additive disharmony functions

Proposition 1 says that the condition (7) that a vector order $\prec$ be additive is phonologically substantive because it ensures that the corresponding CBG $G_\prec$ factorizes. We are thus led to the following question: which assumptions on the disharmony function H suffice to ensure that the corresponding vector order $\prec_H$ defined through (14) satisfies this phonologically substantive additivity condition (7)? We will now see that it suffices to assume that the disharmony of the sum $\mathbf{a} + \mathbf{b}$ of two vectors $\mathbf{a}$ and $\mathbf{b}$ is equal to the sum of their disharmonies, as stated in (15).

$$H(\mathbf{a} + \mathbf{b}) = H(\mathbf{a}) + H(\mathbf{b}) \tag{15}$$

Indeed, let us assume that the numerical order $\prec_H$ induced by a disharmony function H satisfies the antecedent of the additivity implication (7), namely $\mathbf{a} \prec_H \mathbf{b}$. By definition (14), this means in turn that the disharmony $H(\mathbf{a})$ of the smaller vector $\mathbf{a}$ is smaller than the disharmony $H(\mathbf{b})$ of the larger vector $\mathbf{b}$, as in (16a). Let $H(\mathbf{c})$ be the disharmony of the vector $\mathbf{c}$. Whatever this number $H(\mathbf{c})$ looks like, it can be added to both sides of the disharmony inequality $H(\mathbf{a}) < H(\mathbf{b})$ without affecting it, yielding (16b). By the assumption (15) that the disharmony of a sum is the sum of the disharmonies, we can rewrite our inequality as in (16c). Finally, we use again the connection (14) between the disharmony function H and the corresponding numerical order $\prec_H$ to reinterpret the disharmony inequality $H(\mathbf{a} + \mathbf{c}) < H(\mathbf{b} + \mathbf{c})$ as the vector inequality $\mathbf{a} + \mathbf{c} \prec_H \mathbf{b} + \mathbf{c}$ required by the consequent of the additivity implication (7).

$$\begin{aligned} &\mathbf{a} \prec_H \mathbf{b} \iff \\ &\overset{(a)}{\iff} H(\mathbf{a}) < H(\mathbf{b}) \\ &\overset{(b)}{\iff} H(\mathbf{a}) + H(\mathbf{c}) < H(\mathbf{b}) + H(\mathbf{c}) \\ &\overset{(c)}{\iff} H(\mathbf{a} + \mathbf{c}) < H(\mathbf{b} + \mathbf{c}) \\ &\overset{(d)}{\iff} \mathbf{a} + \mathbf{c} \prec_H \mathbf{b} + \mathbf{c} \end{aligned} \tag{16}$$

5.3 Deriving HG's disharmony function

The two preceding subsections have motivated numerical orders defined though disharmony functions which satisfy the identity (15) whereby the disharmony of a sum of vectors is the sum of their disharmonies. We now explore the phonological implications of this assumption (15) by computing the disharmony of the constraint violation vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$ of an arbitrary mapping (x, y) as in (17).

In step (17a), we have recalled that the components of the constraint violation vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$ are the n constraint violations $C_1(\mathsf{x}, \mathsf{y}), \ldots, C_n(\mathsf{x}, \mathsf{y})$. In step (17b), we have baroquely rewritten this constraint violation vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$ as the sum of many vectors: the vector with the 1st component equal to one and all other components equal to zero, repeated $C_1(\mathsf{x}, \mathsf{y})$ times; the vector with the 2nd component equal to one and all other components equal to zero, repeated $C_2(\mathsf{x}, \mathsf{y})$ times; and so on, down to the vector with the nth component equal to one and all other components equal to zero, repeated $C_n(\mathsf{x}, \mathsf{y})$ times.

We now make the crucial assumption that the disharmony function H is additive. This means in particular that the disharmony of a sum of vectors is the sum of their disharmonies (the identity (15) extends trivially from two to an arbitrary finite number of vectors), yielding the identity (17c). Finally, let us call w_1 the disharmony of the vector with the 1st component equal to one and all other components equal to zero; let us called w_2 the disharmony of the vector with the 2nd component equal to one and all other components equal to zero; and so on. The disharmony of the constraint vector $\mathbf{C}(\mathsf{x}, \mathsf{y})$ can thus be described as the sum of the constraint violations $C_1(\mathsf{x}, \mathsf{y}), \ldots, C_n(\mathsf{x}, \mathsf{y})$ rescaled by $w_1, \ldots, w_n$, as stated in (17d).

In conclusion, the reasoning in (17) shows that a disharmony function H that satisfies the additivity condition (15) is the one assumed in HG (Legendre et al. 1990b,a; Smolensky and Legendre 2006). And the HG **weights** $w_1, \ldots, w_n$ can be interpreted as the disharmony of the **base vectors** that have one component equal to one and all other components equal to zero. These base vectors have no phonological meaning (they cannot be interpreted as constraint violation vectors). The reasoning in (17) thus illustrates the advantage of construing CBGs rather abstractly as in section 2, in terms of orders defined among arbitrary vectors.

5.4 The role of constraint integrality

As anticipated in section 2, the constraints used in phonology are assumed to only take (nonnegative) integer values, interpreted as numbers of violations. This assumption formalizes the intuition that the properties relevant to phonology are **discrete**—contrary to the properties relevant to phonetics, which are instead continuous and thus cannot be quantified through just integers. This **constraint integrality assumption** yields a number

$$H(\mathbf{C}(\mathsf{x},\mathsf{y})) \overset{(a)}{=} H\left(\begin{bmatrix} C_1(\mathsf{x},\mathsf{y}) \\ C_2(\mathsf{x},\mathsf{y}) \\ \vdots \\ \vdots \\ C_n(\mathsf{x},\mathsf{y}) \end{bmatrix}\right) \overset{(b)}{=} H\left(\underbrace{\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix} + \ldots + \begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}}_{C_1(\mathsf{x},\mathsf{y})\text{times}} + \underbrace{\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix} + \ldots + \begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}}_{C_2(\mathsf{x},\mathsf{y})\text{ times}} + \ldots + \underbrace{\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix} + \ldots + \begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}}_{C_n(\mathsf{x},\mathsf{y})\text{ times}} \right)$$

$$\overset{(c)}{=} C_1(\mathsf{x},\mathsf{y})\, \underbrace{H\left(\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}\right)}_{w_1} + C_2(\mathsf{x},\mathsf{y})\, \underbrace{H\left(\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}\right)}_{w_2} + \ldots + C_3(\mathsf{x},\mathsf{y}) \underbrace{H\left(\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}\right)}_{w_n}$$

$$\overset{(d)}{=} C_1(\mathsf{x},\mathsf{y})w_1 + C_2(\mathsf{x},\mathsf{y})w_2 + \ldots + C_n(\mathsf{x},\mathsf{y})w_n \tag{17}$$

of finiteness effects when coupled with plausible assumptions on orders among n-dimensional vectors. For instance, Magri (2019) shows that (when coupled with a restriction to vector orders that are monotone), constraint integrality ensures that all candidate sets can be assumed to be finite without loss of generality.

The reasoning in (17) illustrates another finiteness effect of the constraint integrality assumption. Indeed, this reasoning crucially relies on the fact that the constraint violation vector $\mathbf{C}(\mathsf{x},\mathsf{y})$ can be expressed as a sum of a certain number of base vectors. Obviously, this decomposition is only possible because the components $C_k(\mathsf{x},\mathsf{y})$ of a constraint violation vector are integers but would fail otherwise.[1] The reasoning in (17) can thus be interpreted as another finiteness effect of the constraint integrality assumption: when this constraint integrality assumption is coupled with a restriction to numerical orders defined through additive disharmony functions, it ensures that the disharmony function admits a finite representation in terms of a finite number n of weights $w_1, \ldots, w_n$.

6 Conclusions

A phonological representation often factorizes into multiple under-specified representations that each encode only some of the information encoded by the original full-blown representation. We assume that these under-specified representations do not interact, in the sense that the number of constraint violations incurred by a mapping of full-blown representations coincides with the sum of the numbers of constraint violations incurred by the factor mappings of under-specified representations. In this case, we want a phonological grammar that handles full-blown representations to factorize into multiple grammars that handle the under-specified representations independently of each other. This paper has shown that **the HG implementation of constraint-based phonology follows from this factorizability desideratum plus the restriction to disharmony-based orders**. The latter assumption does not seem to admit a phonological justification but it is quite natural from a formal perspective. We conclude that HG admits a principled derivation from axioms that are phonologically or formally motivated (for alternative justifications of the HG framework, see Smolensky and Legendre 2006, and especially chapters 6 and 9). The proposed derivation crucially relies on the constraint-integrality assumption that phonologically relevant properties are discrete. Apart from this constraint-integrality assumption, the reasoning holds without any substantive assumptions on the constraint set.

Acknowledgements

The research reported in this paper has been supported by a JCJC grant from the Agence Nationale de la Recherche (project title: *The mathematics of segmental phonotactics*).

[1] This reasoning (17) is essentially the proof of the Fundamental Theorem of Linear Algebra (Strang 2006), whereby a linear function between finite dimensional spaces admits a matrix representation. The only twist is that we do not need linearity (namely additivity plus homogeneity) but additivity suffices, because we are only dealing with integral vectors.

References

Marlow Anderson and Todd Feil. 1988. *Lattice Ordered Groups: an Introduction*. D. Reidel Publishing Company.

Luigi Burzio. 1996. Surface constraints versus underlying representations. In Jacques Durand and Bernard Laks, editors, *Current trends in Phonology: Models and methods*, pages 97–122. University of Salford Publications, Paris and University of Salford.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990a. Harmonic Grammar: A formal multi-level connectionist theory of linguistic wellformedness: An application. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 884–891, Hillsdale, NJ. Lawrence Erlbaum Associates.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990b. Harmonic Grammar: A formal multi-level connectionist theory of linguistic wellformedness: Theoretical foundations. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 388–395, Hillsdale, NJ. Lawrence Erlbaum.

Giorgio Magri. 2013. A note on the GLA's choice of the current loser from the perspective of factorizability. *Journal of Logic, Language, and Information*, 22:231–247.

Giorgio Magri. 2019. Finiteness of optima in constraint-based phonology. Manuscript, CNRS.

Giorgio Magri and Benjamin Storme. 2020. Constraint summation and the axiomatic derivation of HG. In *Proceedings of the 2019 Annual Meeting on Phonology*. Linguistic Society of America.

Alan Prince. 2015. One tableau suffices. Manuscript (Rutgers University). Available from the Rutgers Optimality Archive as ROA 1250.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in generative grammar*. Blackwell, Oxford. Original version, Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, April 1993. Available from the Rutgers Optimality Archive as ROA 537.

P. Smolensky, M. Goldrick, and D. Mathis. 2014. Optimization and quantization in gradient symbol systems: a framework for integrating the continuous and the discrete in cognition. *Cognitive Science*, 38.6:1102–1138.

Paul Smolensky and Géraldine Legendre. 2006. *The Harmonic Mind*. MIT Press, Cambridge, MA.

Donca Steriade. 1995. Underspecification and markedness. In John Goldsmith, editor, *The handbook of Phonological Theory*, pages 114–174. Blackwell, Oxford.

Gilbert Strang. 2006. *Linear Algebra and its applications*. Thomson Brooks/Cole.

Modeling unsupervised phonetic and phonological learning in Generative Adversarial Phonology

Gašper Beguš
Department of Linguistics, University of Washington
begus@uw.edu

Abstract

This paper models phonetic and phonological learning as a dependency between random space and generated speech data in the Generative Adversarial Neural network architecture and proposes a methodology to uncover the network's internal representation that corresponds to phonetic and phonological features. A Generative Adversarial Network (Goodfellow et al. 2014; implemented as WaveGAN for acoustic data by Donahue et al. 2019) was trained on an allophonic distribution in English, where voiceless stops surface as aspirated word-initially before stressed vowels except if preceded by a sibilant [s]. The network successfully learns the allophonic alternation: the network's generated speech signal contains the conditional distribution of aspiration duration. Additionally, the network generates innovative outputs for which no evidence is available in the training data, suggesting that the network segments continuous speech signal into units that can be productively recombined. The paper also proposes a technique for establishing the network's internal representations. We identify latent variables that directly correspond to presence of [s] in the output. By manipulating these variables, we actively control the presence of [s], its frication amplitude, and spectral shape of the frication noise in the generated outputs.

1 Introduction

Modeling phonetic and phonological data with neural networks has seen a rapid increase in the past few years (Alderete et al. 2013; Avcu et al. 2017; Alderete and Tupper 2018; Mahalunkar and Kelleher 2018; Weber et al. 2018; Dupoux 2018; Prickett et al. 2019; Pater 2019, for cautionary notes, see Rawski and Heinz 2019). The majority of existing computational models in phonology, however, model learning as symbol manipulation and operate with discrete units—either with completely abstract made-up units or with discrete units that feature some phonetic properties that can be approximated as phonemes. This means that either the phonetic and phonological learning are modeled separately or one is assumed to have already been completed with a pre-assumed level of abstraction (Martin et al., 2013; Dupoux, 2018). This is true for both proposals that model phonological distributions or derivations (Alderete et al., 2013; Prickett et al., 2019) and featural organizations (Faruqui et al., 2016; Silfverberg et al., 2018).

Most models in the subset of the proposals that operate with continuous phonetic data assume at least some level of abstraction and operate with already extracted features (e.g. formant values) on limited "toy" data (e.g. Pierrehumbert 2001; Kirby and Sonderegger 2015 for a discussion, see Dupoux 2018). Guenther and Vladusich (2012), Guenther (2016) and Oudeyer (2001, 2002, 2005, 2006), for example, propose models that use simple neural maps that are based on actual correlates of neurons involved in speech production in the human brain (based on various brain imaging techniques). Their models, however, do not operate with raw acoustic data (or require extraction of features in a highly abstract model of articulators; Oudeyer 2005, 2006), require a level of abstraction in the input to the model, and do not model phonological processes — i.e. allophonic distributions. Phonological learning in most of these proposals is thus modeled as if phonetic learning (or at least a subset of phonetic learning) had already taken place: the initial state already includes phonemic inventories, phonemes as discrete units, feature matrices that had already been learned, or extracted phonetic values.

Prominent among the few models that operate with raw phonetic data are Gaussian mixture models for category-learning or phoneme extraction

(Schatz et al., 2019; Lee and Glass, 2012). Schatz et al. (2019) propose a Dirichlet process Gaussian mixture model that learns categories from raw acoustic input in an unsupervised learning task. The primary purpose of the proposal in Schatz et al. (2019) is modeling perception and categorization: they model how a learner is able to categorize raw acoustic data into sets of discrete categorical units that have phonetic values (i.e. phonemes). No phonological processes are modeled in the proposal.

Recently, neural network models for unsupervised feature extraction have seen success in modeling acquisition of phonetic features from raw acoustic data (Kamper et al., 2015). The model in Shain and Elsner (2019), for example, is an autoencoder neural network that is trained on pre-segmented acoustic data. The model takes as an input segmented acoustic data and outputs values that can be correlated to phonological features. Learning is, however, not completely unsupervised as the network is trained on pre-segmented phones. Thiollière et al. (2015) similarly propose an architecture that extracts units from unsupervised speech data. These proposals, however, do not model learning of phonological distributions, but only of feature representations, and crucially are not generative, meaning that the models do not output innovative data, but try to replicate the input as closely as possible (e.g. in the autoencoder architecture).

As argued below, the model based on a Generative Adversarial network learns not only to generate innovative data that closely resemble human speech, but also learns internal representations that resemble phonological features simultaneously with unsupervised phonetic learning from raw acoustic data. Additionally, the model is generative and outputs both the conditional allophonic distributions in the data and innovative data that can be compared to productive outputs in human speech acquisition.

1.1 A Generative Adversarial model of phonology

The advantage of the GAN architecture (Goodfellow et al., 2014; Radford et al., 2015; Donahue et al., 2019) is that learning is completely unsupervised and that phonetic learning is simultaneous with phonological learning in its broadest sense. A network that models learning of phonetics from raw data and shows signs of learning discrete phonological units at the same time is likely one step closer to reality than models that operate with symbolic computation and assume phonetic learning had already taken place and is independent of phonology and vice versa. The Generator's outputs can be approximated as the basis for articulatory targets in human speech that are sent to articulators for execution. The latent variables in the input of the Generator can be modeled as featural representation that the Generator learns to output into a speech signal by attempting to maximize the error rate of a Discriminator network that distinguishes between real data and generated outputs. The Discriminator network thus has a parallel in human speech perception, production, and acquisition: the imitation principle (Nguyen and Delvaux, 2015). The Discriminator's function is to enforce that the Generator's outputs resemble (but not replicate) the inputs as closely as possible. The GAN network thus incorporates both the pre-articulatory production elements (the Generator) as well as the perceptual element (the Discriminator) in speech acquisition. While other neural network architectures might be appropriate for modeling phonetic and phonological learning, GAN is unique in that it is a generative model with the production-perception loop parallel and that, unlike for example autoencoders, generates innovative data rather than data that resembles the input as closely as possible. To our knowledge, this is the first proposal that tests whether neural networks are able to learn an allophonic distribution based on raw acoustic data.

We train a Generative Adversarial Network architecture implemented for audio files in Donahue et al. (2019) (WaveGAN; which is based on DCGAN; Radford et al. 2015) on continuous raw speech data that contains information for an allophonic distribution: word-initial pre-vocalic aspiration of voiceless stops ([ˈpʰɪt] ∼ [ˈspɪt]). The data is curated in order to control for non-desired effects, which is why only sequences of the shape #TV and #sTV (T = stop, V = vowel) are fed to the model. This allophonic distribution is uniquely appropriate for testing learnability in a GAN setting, because the dependency between the presence of [s] and duration of VOT is not strictly local. To be sure, the dependency is local in phonological terms, as [s] and T are two segments and immediate neighbors, but in phonetic terms, a period of closure intervenes between the aspiration and the period (or absence thereof) of frication noise of [s].

The hypothesis of the computational experiment presented in Section 3 is the following: if VOT duration is conditioned on the presence of [s] in output data generated from noise by the Generator network, it means that the Generator network has successfully learned a phonetically non-local allophonic distribution. Because the allophonic distribution is not strictly local and not automatic, but has to be learned and actively controlled by speakers, evidence for this type of learning is considered phonological learning in the broadest sense. Conditioning the presence of a phonetic feature based on the presence or absence of a phoneme that is not automatic is, in most models, considered part of phonology and is derived with phonological computation. That the tested distribution is non-automatic and has to be actively controlled by the speakers is evident from L1 acquisition: failure to learn the distribution results in longer VOT durations in the sT condition documented in L1 acquisition (McLeod et al., 1996; Bond, 1981). Additional evidence that the GAN's learning resembles phonemic representations (such as presence of [s]) is obtained from recovering the networks' internal representations (see below and Section 3.2).

This paper also proposes a technique for establishing the Generator's internal representations. What neural networks actually learn is a challenging question with no easy solutions. The inability to uncover networks' representations has been used as an argument against neural network approaches to linguistic data (Rawski and Heinz, 2019). We argue that internal representation of a network can be, at least partially, uncovered. By regressing annotated dependencies between the Generator's latent space and output data, we identify values in the latent space that correspond to linguistically meaningful features in generated outputs. This paper demonstrates that manipulating the chosen values in the latent space have phonetic and phonological effects in the generated outputs, such as the presence of [s] and the amplitude of its frication. In other words, the GAN network learns to use random noise as an approximation of phonetic and phonological features. This paper proposes that dependencies, learned during training in a latent space that is limited by some interval, extend beyond that interval. This crucial step allows for the discovery of several phonetic properties.

2 Materials

2.1 The model: Donahue et al. (2019) based on Radford et al. (2015)

Generative Adversarial Networks, proposed by Goodfellow et al. (2014), have seen a rapid expansion in a variety of tasks, including but not limited to computer vision and image generation (Radford et al., 2015). The main characteristic of GANs is the architecture that involves two networks: the Generator network and the Discriminator network (Goodfellow et al., 2014). The Generator network is trained to generate data from random noise, while the Discriminator is trained to distinguish real data from the outputs of the Generator network (Figure 1). The Generator is trained to generate data that maximizes the error rate of the Discriminator network. The training results in a Generator (G) network that takes random noise as its input (e.g. multiple variables with uniform distributions) and outputs data such that the Discriminator is inaccurate in distinguishing the generated from the real data.

Applying the GAN architecture on time-series data such as a continuous speech stream faces several challenges. Recently, Donahue et al. (2019) proposed an implementation of a Deep Convolutional Generative Adversarial Network proposed by Radford et al. (2015) for audio data (WaveGAN); the model along with the code in Donahue et al. (2019) was used for training in this paper. The model takes one-second long raw audio files as inputs, sampled at 16 kHz with 16-bit quantization. The audio files are converted into a vector and fed to the Discriminator network as real data. Instead of the two-dimensional 5×5 filters, the WaveGAN model uses one-dimensional 1×25 filters and larger upsampling (Donahue et al., 2019). The main architecture is preserved as in DCGAN, except that an additional layer is introduced in order to generate longer samples. The Generator network takes as input z, a vector of one hundred uniformly distributed variables ($z \sim \mathscr{U}(-1,1)$) and outputs 16,384 data points, which constitutes the output audio signal. The network has five 1D convolutional layers (Donahue et al., 2019). The Discriminator network takes 16,384 data points (raw audio files) as its input and outputs a sin-

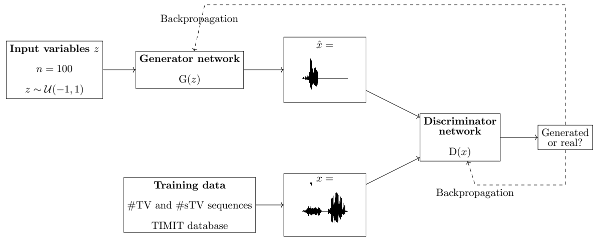


Figure 1: A diagram showing the Generative Adversarial architecture as proposed in Goodfellow et al. (2014); Donahue et al. (2019) and trained on data from the TIMIT database in this paper.

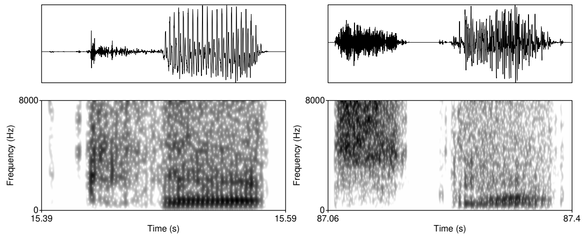


Figure 2: Waveforms and spectrograms (0–8,000 Hz) of a typical generated samples of #TV (left) and #sTV (right) sequences from a Generator trained after 12,255 steps.

gle logit. The initial GAN design as proposed by Goodfellow et al. (2014) trained the Discriminator network to distinguish real from generated data. Training such models, however, faced substantial challenges (Donahue et al., 2019). Donahue et al. (2019) implement the WGAN-GP strategy (Arjovsky et al., 2017; Gulrajani et al., 2017), which means that the Discriminator is trained "as a function that assists in computing the Wasserstein distance" (Donahue et al., 2019). The WaveGAN model (Donahue et al., 2019) uses ReLU activation in all but the last layer for the Generator network, and Leaky ReLU in all layers in the Discriminator network (as recommended for DCGAN in Radford et al. 2015). For exact dimensions of each layer and other details of the model, see Donahue et al. (2019).

2.2 Training data

The model was trained on the allophonic distribution of voiceless stops in English. Voiceless stops /p, t, k/ surface as aspirated [p^{h}, t^{h}, k^{h}] in English in word-initial position when immediately followed by a stressed vowel (Lisker, 1984; Iverson and Salmons, 1995; Vaux, 2002; Vaux and Samuels, 2005; Davis and Cho, 2006). If an alveolar sibilant [s] precedes the stop, however, the aspiration is blocked and the stop surfaces as unaspirated [p, t, k] (Lisker, 1984). A minimal pair illustrating this allophonic distribution is [ˈp^{h}ɪt] 'pit' vs. [ˈspɪt] 'spit'. The most prominent phonetic correlate of this allophonic distribution is the difference in Voice Onset Time (VOT) duration (Abramson and Whalen, 2017) between the aspirated and unaspirated voiceless stops.

The model was trained on data from the TIMIT database (Garofolo et al., 1993).[1] The training data consist of 16-bit .wav files with 16 kHz sampling rate of word initial sequences of voiceless stops /p, t, k/ (= T) that were followed by a vowel (#TV) and word initial sequences of /s/ + /p, t, k/, followed by a vowel (#sTV). The training data includes 4,930 sequences with the structure #TV and 533 sequences with the structure #sTV (5,463 total). Both stressed and unstressed vowels are included in the training data, as this condition crucially complicates learning and makes the task for the neural network more challenging.

3 Experiment

3.1 Model: 12,255 steps

The Generator network after 12,255 steps ($\sim$ 716 epochs) generates an acoustic signal that appears close to actual speech data. The number of training steps was chosen manually as a compromise between output interpretability and the number of epochs, where we try to approximately maximize the first and minimize the latter parameter. Figure 2 illustrates a typical generated sample of #TV (left) and #sTV (right) structures with a substantial difference in VOT durations.

To test whether the Generator learns the conditional distribution of VOT duration, the generated samples were annotated for VOT duration. VOT duration was measured from the release of closure to the onset of periodic vibration with clear formant structure. Altogether 96 generated samples were annotated; 62 in which no period of frication of [s] preceded and 34 in which [s] precedes the TV sequence. The generated data were fit to a linear model with only one predictor: presence of [s] (STRUCTURE). Place of articulation or fol-

[1] Donahue et al. (2019) trained the model on the SC09 and TIMIT databases, but the results are not useful for modeling phonological learning, because the model is trained on a continuous speech stream and the generated sample fails to produce analyzable results for phonological purposes.

lowing vowel were not added in the model, because they are often difficult to recover. STRUCTURE is a significant predictor of VOT duration: $F(1) = 53.1, p < 0.0001$. The estimates for Intercept (duration of VOT when no [s] precedes) are $\beta = 56.2$ ms$, t = 25.74, p < 0.0001$. VOT is on average 26.8 ms shorter if [s] precedes the TV sequence and this difference is significant ($\beta = -26.8$ ms$, t = -7.29, p < 0.0001$).

While VOT duration is significantly shorter if [s] precedes the #TV sequence in the generated data, the model shows clear traces that the learning is incomplete and that the generator network fails to learn the distribution *categorically* at 12,255 steps. The three longest VOT durations in the #sTV condition in the generated data are 68.3 ms, 75.7 ms, and 76.2 ms. In all three cases the VOT is longer than the longest VOT duration of any #sTV sequence in the training data (longest is 65 ms). This generalization holds even in proportional terms (i.e. while controlling for "speech rate"): the generated data contains the highest ratio between the VOT duration and the frication duration of [s].

Longer VOT duration in the #sTV condition in the generated data compared to training data is not the only violation of the training data that the Generator outputs and that resembles linguistic behavior in humans. Occasionally, the Generator outputs a linguistically valid #sV sequence for which no evidence was available in the training data. The minimal duration of closure in #sTV sequences in the training data is 9.2 ms, the minimal duration of VOT is 9.4 ms. All sequences containing a [s] from the training data were manually inspected, and none of them contain a #sV sequence without a period of closure and VOT. Homorganic sequences of [s] followed by an alveolar stop [t] (#stV) are occasionally acoustically similar to the sequence without the stop (#sV) because frication noise from [s] carries onto the homorganic alveolar closure which can be very short. However, there is a clear fall and a second rise of noise amplitude after the release of the stop in #stV sequences. Figure 3 shows one case of the Generator network outputting a #sV sequence without any stop-like fall of the amplitude. In other words, the Generator network outputs a linguistically valid sequence #sV without any evidence for existence of this sequence in the training data. Similarly, the Generator occasionally outputs a sequence with two stops (two periods of aspiration noise with intervening short period of closure) and a vowel (#TTV) (Figure 3).

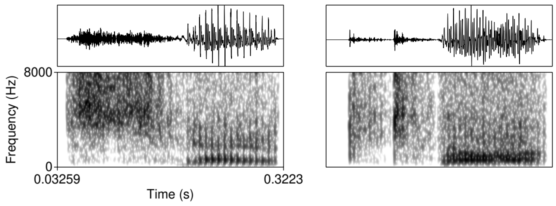


Figure 3: Waveforms and spectrograms (0–8000 Hz) of two innovative generated outputs of the shape #sV and #TTV. The sample on the left was generated after 16,715 steps.

Measuring overfitting is a substantial problem for Generative Adversarial Networks with no consensus on the most appropriate quantitative approach to the problem (Goodfellow et al., 2014; Radford et al., 2015). The danger with overfitting in a GAN architecture is that the Generator network would learn to fully replicate the input. Donahue et al. (2019) test overfitting on models trained with a substantially higher number of steps (200,000) compared to our model (12,255) and presents evidence that GAN models trained on audio data do not overfit even with substantially higher number of training steps. The best evidence against overfitting is precisely the fact that the Generator network outputs samples that substantially violate output distributions.

3.2 Establishing internal representations

Establishing internal representations of a neural network is a challenging task (Lillicrap and Kording, 2019). Below, we propose a technique for uncovering dependencies between the network's latent space and generated data based on logistic regression. This method has the potential to shed light on the network's internal representations: using the proposed technique, we can estimate how the network learns to map latent space into phonetically and phonologically meaningful units in the generated data.

To identify dependencies between the latent space and generated data, we correlate annotations of the output data with the variables in the latent space. As a starting point, we choose to identify correlates of the most prominent feature in the training data: presence or absence of [s]. Any number of other phonetic features can be correlated with this approach; applying this technique to other features and other alternations should yield a better understanding of the network's learning mechanisms. Focusing on more than the chosen feature, however, is beyond the scope of this paper.

We propose a method based on logistic regression. First, 3,800 outputs from the Generator network trained after 12,255 steps were generated and manually annotated for presence or absence of [s]. 271 outputs (7.13%) were annotated as involving a segment [s]. Frication that resembled [s]-like aspiration noise after the alveolar stop and before high vowels was not annotated as including [s]. Innovative outputs such as an #[s] without the following vowel or #sV sequences were annotated as including an [s].

The annotated data together with values of latent variables for each generated sample (z) were fit to a logistic regression generalized additive model (using the *mgcv* package; Wood 2011 in R Core Team 2018) with the presence or absence of [s] as the dependent variable (binomial distribution of successes and failures) and smooth terms of latent variables (z) as predictors of interest (estimated as penalized thin plate regression splines; Wood 2011). Generalized additive models were chosen in order to avoid assumptions of linearity: it is possible that latent variables are not linearly correlated with features of interest in the output of the Generator network. The initial full model (FULL) includes smooths for all 100 variables in the latent space that are uniformly distributed within the interval $(-1, 1)$ as predictors.

To reduce the number of variables, models with different shrinkage techniques are refit and compared: the latent variables for further analysis are then chosen based on combined results of different extratory models. We refit the model with various modifications: with modified smoothing penalty (MODIFIED); with original smoothing penalty, but with an additional penalty for each term if all smoothing parameters tend to infinity (SELECT; Wood 2011); and with manual removal of non-significant terms by Wald test for each term (EXCLUDED).

The estimated smooths appear mostly linear. We also fit the data to a linear logistic regression model (LINEAR) with all 100 predictors. To reduce the number of predictors, another model is fit (LINEAR EXCLUDED) with those predictors removed that do not improve fit.

To identify latent variables with highest correlation with [s] in the output, we extract estimates for each term from the generalized additive models and estimates of slopes from the linear model. Figure 4 plots those values. The plot points to a substantial difference between the highest seven predictors and the rest of the latent space. Seven latent variables are thus identified (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) as potentially having the largest effect on presence or absence of [s] in output. Lasso regression (Simon et al., 2011) and Random Forest models (Liaw and Wiener, 2002) give almost identical results.

To conduct an independent generative test of whether the chosen values correlate with [s] in the output data of the Generator network, we set values of the seven identified predictors (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) to the marginal value of 1 or -1 (depending on whether the correlation is positive or negative) and generated 100 outputs. Altogether seven values in the latent space were thus manipulated, which represents only 7% of the entire latent space. Of the 100 outputs with manipulated values, 73 outputs included a [s] or [s]-like element, either with the stop closure and vowel or without them. The rate of outputs that contain [s] is thus significantly higher when the seven values are manipulated to the marginal levels compared to randomly chosen latent space. In the output data without manipulated values, only 271 out of 3800 generated outputs (or 7.13%) contained an [s]. The difference is significant ($\chi^2(1) = 559.0, p < 0.00001$).

High proportions of [s] in the output can be achieved with manipulation of single latent variables, but the values need to be highly marginal, i.e. extend well beyond the training space. Setting the z_{11} value outside the training interval to -15, for example, causes the Generator to output [s] in 87 out of 100 generated (87%) sequences, which is again significantly more than with random input ($\chi^2(1) = 792.7, p < 0.0001$). When z_{11} is -25, the rate goes up to 96 out of 100, also significantly different from random inputs ($\chi^2(1) = 959.8, p < 0.0001$).

While there is a consistent drop in estimates of the regression models after the seven identified variables (Figure 4) and while several independent generation tests confirm that the seven variables correspond the to presence of [s] in the output,

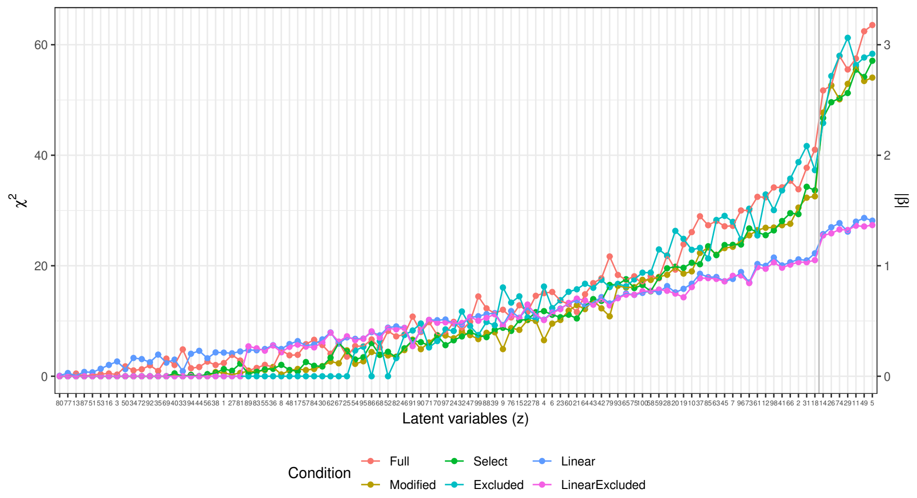


Figure 4: Plot of χ^2 values (left scale) for the 100 predictors across the four generalized additive models. For the two linear models (LINEAR and LINEAR EXCLUDED), estimates of slopes in absolute values ($|\beta|$) are plotted (right scale). The blue vertical line indicates the division between the seven chosen predictors and the rest of the predictor space with a clear drop in estimates between the first seven values (z_5, z_{11}, z_{49}, z_{29}, z_{74}, z_{26}, z_{14}) and the rest of the space.

the cutoff point between the seven variables and the rest of the latent space is still somewhat arbitrary. It is likely that other latent variables directly or indirectly influence the presence of [s] as well: the learning at this point is not yet categorical and several dependencies not discovered here likely affect the results. Nevertheless, further explorations of the latent space suggest the variables identified with the logistic regression (and other) models (Figure 4) are indeed the main variables involved with the presence or absence of [s] in the output.

3.3 Interpolation and phonetic features

We further explore whether the mapping between the uniformly distributed input (z) variables can be associated with specific phonetic or phonological features in that output. The crucial step in this direction is to explore values of the latent space beyond the training interval, i.e. beyond $(-1,1)$. Crucially, we observe that the Generator network, while being trained on latent space limited to the interval $(-1,1)$, learns representations that extend this interval. Even if the input latent variables (z) exceed the training interval, the Generator network outputs samples that closely resemble human speech. Furthermore, the dependencies learned during training extend outside of the $(-1,1)$ interval. Exploring phonetic properties at these marginal values might reveal the actual underlying function of each latent variable.

To explore phonetic correlates of the seven latent variables, we set each of the seven variables separately to the marginal value -4.5 and interpolate to its opposite marginal value 4.5 in 0.5 increments, while keeping randomly-sampled values of the other 99 latent variables z constant. The ± 4.5 value was chosen based on manual inspection of generated samples: amplitude rises of [s] gradually weaken when variables have a value greater than ± 3.5. Seven sets of generated samples are thus created, one for each of the seven z values (with the other 99 z-values randomly sampled, but kept constant for all seven manipulated variables). Each set contains a subset of 19 generated outputs that correspond to the interpolated variables from -4.5 to 4.5 in 0.5 increments. Twenty-nine such sets containing an [s] in at least one set are extracted for analysis.

A clear pattern emerges in the generated data: the latent variables identified as corresponding to the presence of [s] via regression (Figure 4) have direct phonetic correlates and cause changes in amplitude and presence/absence of frication noise of [s] when each of the seven values in the latent space are manipulated to the chosen values, including values that exceed the training interval. In other words, by manipulating the identified latent variables, we control the presence/absence of [s] in the output as well as the amplitude of its frication noise.

Figure 5 illustrates this effect. Frication noise of

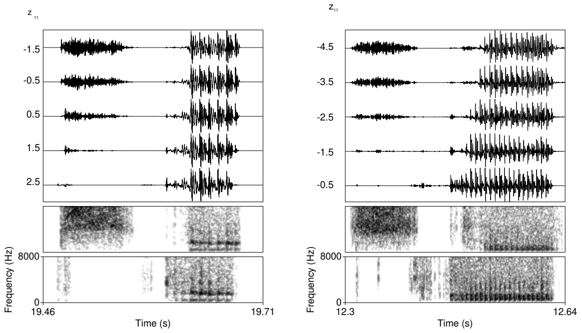


Figure 5: Waveforms and two spectrograms (both 0 – 8,000 Hz) of generated data with z_{11} variable manipulated and interpolated. The values on the left of waveforms indicate the value of z_{11}. The two spectrograms represent the highest and the lowest value of z_{11}. A clear attenuation of the frication noise is visible until complete disappearance.

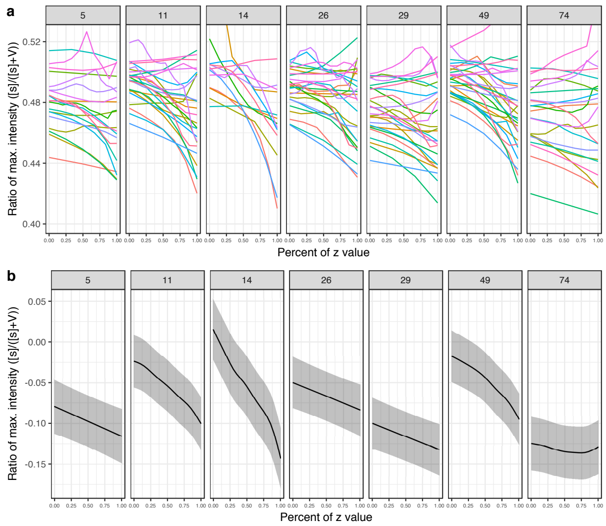


Figure 6: **(a)** Plots of ratios of maximum intensity between the frication of [s] and phonation of the vowel in #sTV sequences across the seven variables and **(b)** predicted values with 95% CIs of the ratio based on beta regression generalized additive model.

[s] gradually decreases by increasing the value of z_{11} until it completely disappears. The exact value of z_{11} for which the [s] disappears differs across examples and likely interacts with other features. It is possible that frication noise in the training has a higher amplitude in some conditions, which is why such cases require a higher magnitude of manipulation of z_{11}. The figure also shows that as the frication noise of [s] disappears, aspiration of a stop in what appears to be a #TV sequences starts surfacing and replaces the frication noise of [s]. Occasionally, frication noise of [s] gradually transforms into aspiration noise. The exact transformation is likely dependent on the 99 other z-variables held constant and their underlying phonetic effect. Regardless of the underlying phonetic effect of the other variables in the latent space, we can force [s] in the output when generating data and manipulating the chosen variables.

To test the significance of the effects of the seven identified features on the presence of [s] and the amplitude of its frication noise, the 29 generated sets of 19 outputs (with z-value from -4.5 to 4.5) for each of the seven variables were analyzed. The outputs were manually annotated for [s] and the following vowel. Outputs gradually change from #sTV to #TV. Only sequences containing an [s] were analyzed; as soon as [s] stops in the output, annotations were stopped and the outputs were not further analyzed. For each data point, maximum intensity of the fricative and the vowel was extracted in Praat (Boersma and Weenink, 2015; Lennes, 2003) with a 13.3 ms window length.

To test whether the decreased frication noise is not part of a general effect of decreased amplitude, we perform significance tests on the ratio of maximum intensity between the frication noise of [s] and the following vowel in the #sTV sequences. Figure 6 plots the ratio of maximum intensity of the fricative divided by the sum of two maximum intensities: of the fricative ([s]) and of the vowel (V). The manipulated z-values are additionally normalized to the interval [0,1], where 0 represents the most marginal value with [s] (usually ±4.5; referred to as STRONG henceforth) and 1 represents the last value before [s] disappears (WEAK). Note that the point at which [s] is not present in the output anymore, but the vowel still surfaces (which would yield the ratio at 0) is not included in the model.

The data were fit to a beta regression generalized additive mixed model (Wood 2011) with random smooths for (i) trajectory and for (ii) value of other variables in the latent space of the Generator network, see Figure 6. All smooths (except for z_{74}) are significantly different from 0 and the plots show a clear negative trajectory.

The seven variables thus strongly correspond to the presence or absence of [s] in the output; by manipulating the chosen variables to the identified values we can attenuate frication noise of [s] and cause its presence or complete disappearance in the generated data. Again, the discovery of these features is possible because we extend the initial training interval and test predictions on marginal values.

Interpolation of latent variables reveals that the presence of [s] is not controlled by a single latent variable, but by at least seven of them. The different latent variables that correspond to the presence of [s], however, are not phonetically vacuous: individually, they have distinct phonetic correspondences. The generated samples reveal that the variables' secondary effect (besides outputting [s] and controlling its intensity) is likely reflected in spectral properties of the frication noise. The seven variables are thus similar in the sense that manipulation of their values results in the presence of [s] by controlling its frication noise. They crucially differ, however, in the effects on the spectral properties of the outputs.

To test this prediction, spectral properties of the output fricatives are analyzed in the same 29 sets of generated samples. Spectral properties of the generated fricatives are generally not significantly different at the value of z right before [s] disappears from the outputs. As values of z increase toward the marginal levels (in most cases, ± 4.5), however, clear differentiation in spectral properties emerge between the seven z-variables. The trajectory for center of gravity, for example, significantly differs between z_{11} and most of the other six variables. Overall kurtosis is significantly different when z_{11} is manipulated, compared to, for example, z_{26} and z_{29}. Similarly, while z_{74} does not significantly attenuate amplitude of [s], it significantly differs in skew trajectory of [s]. The main function of z_{74} is thus likely in its control of spectral properties of frication of [s] (e.g. skew).

In sum, manipulating the latent variables that correspond to [s] in the output not only attenuates frication noise (when vocalic amplitude is controlled for) and causes [s] to surface or disappear from the output, but the different z-variables likely correspond to different phonetic features of the frication noise. By setting the values to the marginal levels well beyond the training interval, however, significant differences emerge both in overall levels as well as in trajectories of COG, kurtosis, and skew. It is thus likely that the variables collectively control the presence or absence of [s], but that individually, they control various phonetic features — spectral properties of the frication noise.

4 Conclusion

The results of this paper suggest that we can model phonology not only with rules (Chomsky and Halle, 1968), finite-state automata (Heinz, 2010; Chandlee, 2014), input-output optimization (Prince and Smolensky, 1993/2004), or with neural network architecture that already assumes some level of abstraction (see Section 1), but as the dependency between the latent space and generated data in Generative Adversarial Networks that are trained in an unsupervised manner from raw acoustic data. We train a Generative Adversarial Network (as implemented in Donahue et al. 2019 based on DCGAN architecture; Radford et al. 2015); the results of the computational experiment suggest that the network learns the conditional allophonic distribution of VOT duration. To the author's knowledge, this is the first paper testing learning of allophonic distributions in an unsupervised manner from raw acoustic data using neural networks. This paper also proposes a technique that identifies variables that correspond to the presence of [s] in the output and shows that by manipulating these values, we can generate data with or without [s] in the output as well as control its intensity and spectral properties of its frication noise. While at least seven latent variables control the presence of [s], each of them has a phonetic function that controls spectral properties of the frication noise. The proposed technique thus suggests that the Generator network learns to encode phonetic and phonological information in its latent space.

Training GAN networks on further processes and on languages other than English should yield more information about learning representations of phonetic and phonological processes. This paper outlines methodology for establishing internal representations and testing predictions against generated data, but represents just a first step in a broader task of establishing learning representation of phonetic and phonological data in a Generative Adversarial framework of phonology.

Acknowledgments

This research was funded by a grant to new faculty at the University of Washington. I would like to thank Sameer Arshad for slicing data from the TIMIT database and Heather Morrison for annotating data. All mistakes are my own.

References

Arthur S. Abramson and D.H. Whalen. 2017. Voice onset time (vot) at 50: Theoretical and practical issues in measuring voicing distinctions. *Journal of Phonetics*, 63:75 – 86.

John Alderete and Paul Tupper. 2018. Connectionist approaches to generative phonology. In Anna Bosch and S. J. Hannahs, editors, *The Routledge Handbook of Phonological Theory*, pages 360–390. Routledge, New York.

John Alderete, Paul Tupper, and Stefan A. Frisch. 2013. Phonological constraint induction in a connectionist network: learning ocp-place constraints from data. *Language Sciences*, 37:52 – 69.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

Enes Avcu, Chihiro Shibata, and Jeffrey Heinz. 2017. Subregular complexity and deep learning. In *Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML)*.

Paul Boersma and David Weenink. 2015. Praat: doing phonetics by computer [computer program]. version 5.4.06. Retrieved 21 February 2015 from http://www.praat.org/.

Z. S. Bond. 1981. A note concerning /s/ plus stop clusters in the speech of language-delayed children. *Applied Psycholinguistics*, 2(1):55–63.

Jane Chandlee. 2014. *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Stuart Davis and Mi-Hui Cho. 2006. The distribution of aspirated stops and /h/ in American English and Korean: an alignment approach with typological implications. *Linguistic*, 41(4):607–652.

Chris Donahue, Julian McAuley, and Miller Puckette. 2019. Adversarial audio synthesis. In *ICLR*. `github.com/chrisdonahue/wavegan`.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43 – 59.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California. Association for Computational Linguistics.

J. S. Garofolo, Lori Lamel, W M Fisher, Jonathan Fiscus, D S. Pallett, N L. Dahlgren, and V Zue. 1993. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Frank H Guenther. 2016. *Neural control of speech*. MIT Press.

Frank H. Guenther and Tony Vladusich. 2012. A neural theory of speech acquisition and production. *Journal of Neurolinguistics*, 25(5):408 – 422. Is a neural theory of language possible? Issues from an interdisciplinary perspective.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc.

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Gregory K. Iverson and Joseph C. Salmons. 1995. Aspiration and laryngeal representation in germanic. *Phonology*, 12(3):369–396.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5818–5822.

James Kirby and Morgan Sonderegger. 2015. Bias and population structure in the actuation of sound change. *arXiv e-prints*, page arXiv:1507.04420.

Chia-ying Lee and James Glass. 2012. A nonparametric Bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea. Association for Computational Linguistics.

Mietta Lennes. 2003. f0-f1-f2-intensity_praat_script. praat script. Modified by Dan McCloy, Esther Le Grésauze, and Gašper Beguš.

Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomforest. *R News*, 2(3):18–22.

Timothy P. Lillicrap and Konrad P. Kording. 2019. What does it mean to understand a neural network? *arXiv e-prints*, page arXiv:1907.06374.

Leigh Lisker. 1984. How is the aspiration of english /p, t, k/ "predictable"? *Language and Speech*, 27(4):391–394.

Abhijit Mahalunkar and John D. Kelleher. 2018. Using regular languages to explore the representational capacity of recurrent neural architectures. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 189–198, Cham. Springer International Publishing.

Andrew Martin, Sharon Peperkamp, and Emmanuel Dupoux. 2013. Learning phonemes with a proto-lexicon. *Cognitive Science*, 37(1):103–124.

S McLeod, J van Doorn, and V Reed. 1996. Homonyms and cluster reduction in the normal development of children's speech. In *Proceedings of the Sixth Australian International Conference on Speech Science & Technology*, pages 331–336.

Noël Nguyen and Véronique Delvaux. 2015. Role of imitation in the emergence of phonological systems. *Journal of Phonetics*, 53:46 – 54. On the cognitive nature of speech sound systems.

Pierre-Yves Oudeyer. 2001. Coupled neural maps for the origins of vowel systems. In *Proceedings of the International conference on artificial neural networks. Lecture notes in computer science*, pages 1171–1176. Springer. Volume: 2130.

Pierre-Yves Oudeyer. 2002. Phonemic coding might result from sensory-motor coupling dynamics. In *From animals to animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 406–416. MIT Press.

Pierre-Yves Oudeyer. 2005. The self-organization of speech sounds. *Journal of Theoretical Biology*, 233(3):435 – 449.

Pierre-Yves Oudeyer. 2006. *Self-organization in the evolution of speech*. Studies in the evolution of language ; 6. Oxford University Press, Oxford.

Joe Pater. 2019. Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language*.

Janet Pierrehumbert. 2001. Exemplar dynamics: Word frequency, lenition, and contrast. In Joan L. Bybee and Paul J. Hopper, editors, *Frequency Effects and the Emergence of Lexical Structure*, pages 137–157. John Benjamins, Amsterdam.

Brandon Prickett, Aaron Traylor, and Joe Pater. 2019. Learning reduplication with a variable-free neural network. Ms., University of Massachusetts, Amherst. http://works.bepress.com/joe_pater/38/ (accessed 23 May 2019).

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell, Malden, MA. First published in Tech. Rep. 2, Rutgers University Center for Cognitive Science.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Jonathan Rawski and Jeffrey Heinz. 2019. No free lunch in linguistics or machine learning: Response to pater. *Language*.

Thomas Schatz, Naomi Feldman, Sharon Goldwater, Xuan Nga Cao, and Emmanuel Dupoux. 2019. Early phonetic learning without phonetic categories – insights from machine learning.

Cory Shain and Micha Elsner. 2019. Measuring the perceptual availability of phonological features during language acquisition using unsupervised binary stochastic autoencoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 69–85, Minneapolis, Minnesota. Association for Computational Linguistics.

Miikka P. Silfverberg, Lingshuang Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2011. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13.

Roland Thiollière, Ewan Dunbar, Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux. 2015. A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In *Proceedings of Interspeech*.

Bert Vaux. 2002. Aspiration in English. Ms., Harvard University. Accessed on June 27, 2019.

Bert Vaux and Bridget Samuels. 2005. Laryngeal markedness and aspiration. *Phonology*, 22(3):395–436.

Noah Weber, Leena Shekhar, and Niranjan Balasubramanian. 2018. The fine line between linguistic generalization and failure in Seq2Seq-attention models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, New Orleans, Louisiana. Association for Computational Linguistics.

S. N. Wood. 2011. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.

Phonotactic learning with neural language models

Connor Mayer
Department of Linguistics
University of California, Los Angeles
connormayer@ucla.edu

Max Nelson
Department of Linguistics
University of Massachusetts, Amherst
manelson@umass.edu

Abstract

Computational models of phonotactics share much in common with language models, which assign probabilities to sequences of words. While state of the art language models are implemented using neural networks, phonotactic models have not followed suit. We present several neural models of phonotactics, and show that they perform favorably when compared to existing models. In addition, they provide useful insights into the role of representations on phonotactic learning and generalization. This work provides a promising starting point for future modeling of human phonotactic knowledge.

1 Introduction and background

1.1 Phonotactics

Research on phonotactics deals broadly with two questions: what kinds of knowledge do speakers have about about the phonotactics of their language, and how is this knowledge acquired? (e.g., Chomsky and Halle, 1965) One important outcome of this work has been to show that phonotactic judgements are not categorical, but exhibit gradience: i.e., some possible words are better than others. For example, while /wɪs/ and /ploʊmf/ are both judged as being possible English words by speakers, the former is consistently judged to be a 'better' English word than the latter (Albright and Hayes, 2003; Albright, 2009). Phonotactic modelling studies have tried to build computational models of phonotactic knowledge that agree with gradient human phonotactic judgements. These models provide insight into the structure of phonological knowledge, which aspects of the data are considered by the learner when constructing their phonological grammar, and what biases constrain the forms these grammars may take (e.g., Hayes and Wilson, 2008; Albright, 2009; Daland et al., 2011; Futrell et al., 2017; Jarosz and Rysling, 2017).

1.2 Phonotactics and language modeling

The task undertaken by models of phonotactics is similar in many respects to the more general task of language modeling. A language model assigns probabilities to sequences of words, defining a probability distribution over word sequences (e.g., Jurafsky and Martin, 2008). A simple form of language modeling calculates n-gram probabilities based on corpus frequencies, and uses these to assign probabilities to longer sequences.

Phonotactic models, and models of related tasks such as word segmentation (e.g., Schrimpf and Jarosz, 2014), often frame the problem as one of language modeling over sounds rather than words. They attempt to assign probabilities to phoneme sequences that distinguish licit and illicit forms, correspond to gradient human judgements, or facilitate some task such as word segmentation. These models almost invariably operate on some version of n-grams, though they differ in whether they consider segments (e.g., Jelinek, 1999; Vitevitch and Luce, 2004; Jurafsky and Martin, 2008), phonological features (e.g., Albright, 2009), combinations of the two (e.g., Albright, 2009; Futrell et al., 2017), or larger prosodic structures (e.g., Coleman and Pierrehumbert, 1997; Yang, 2004; Swingley, 2005; Phillips and Pearl, 2015) to be the primitives from which sequences are built.

While early language models relied on the same types of variations on the n-gram employed by phonotactic learners, language modeling in NLP has seen a shift away from count-based, parametric n-gram models. Bengio et al. (2003) introduced a neural n-gram model which still makes predictions based on a fixed-size history window, but uses a neural network to generate the probability function from the history rather than simple

n-gram counts. Bengio et al. (2003) also introduced the idea of learning word embeddings while optimizing for the language modeling task: vector representations of words that are determined based on the word's distribution in the training data.

One shortcoming of n-gram models, neural or otherwise, is that the context window is fixed and specified by the researcher. This is particularly problematic for cases in which long-distance dependencies are numerous and can operate over arbitrary distances. To mitigate this issue, Mikolov et al. (2010) introduced Recurrent Neural Network Language Models (RNNLMs). These networks make use of recurrent connections to store information over potentially unbounded distances.[1] The idea of training recurrent networks on next element prediction dates to the introduction of RNNs in Elman (1990), where RNNs trained on next letter prediction were shown to learn simple phonotactic patterns like CV alternation.

Part of what the RNNLM learns is what information in the history should be considered when processing the current word. In this way RNNLMs trained on a language modeling objective are able to base predictions on all preceding information rather than just the previous n words.

The RNNLM and its descendants, including LSTM language models (Sundermeyer et al., 2012) and deep contextual language models (Peters et al., 2018), have yielded dramatic improvements in performance on language modeling benchmarks, but have seen little application as phonotactic models until recently. Silfverberg et al. (2018) show that phoneme representations learned with neural methods developed for word embeddings (Word2Vec) cluster in ways that correspond to phonetic properties, and can be used to predict sound analogies. Mirea and Bicknell (2019), in a recent application of the language modeling objective to phonotactic learning, train LSTM language models on an English lexicon, and demonstrate the potential value of neural LMs as phonotactic learners.

1.3 The goals of this paper

The primary goal of this paper is to show that relatively simple neural network architectures developed for language modeling can be easily adapted to serve as phonotactic models, and that these models perform favorably when compared to existing models. In addition, we will show that the adoption of these neural models allows theoretical predictions about the role of representations in phonotactic grammars to be tested in ways that are not straightforward with existing models. We will demonstrate this on three phonemic data sets that exhibit phonotactic properties that have proven interesting or challenging for past models of phonotactics, and for phonological theory in general.

[1] Though in practice RNNs cannot capture arbitrarily long-distance dependencies (Bengio et al., 1994).

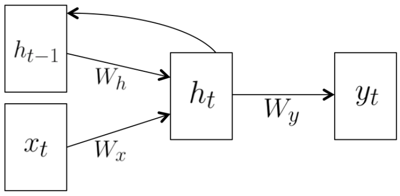


Figure 1: Schematic sRNN architecture

2 Model architectures

The RNNLM for phonotactic learning aims to define a probability distribution over upcoming phonemes given a representation of all preceding phonemes. We will focus on Simple Recurrent Neural Network (sRNN) variants of the models (Elman, 1990). sRNNs are a type of RNN in which the network's state at any timepoint is dependent only on the current input and the network's state at the immediately preceding timepoint (Fig. 1). The computation of the vector representing the network's state at time t, h_t, is shown in (1).

$$h_t = tanh(W_x x_t + W_h h_{t-1} + b_h) \qquad (1)$$

x_t is the embedding vector corresponding to the phoneme input at time t, W_x and W_h are weight matrices for the input and previous state vectors respectively, and b_h is a bias vector. h_t is then used to produce a probability distribution over phonemes, $\hat{y}_t$, which is the model's prediction of the identity of the segment that will appear at time $t+1$. $\hat{y}_t$ is calculated as

$$\hat{y}_t = \sigma(W_y h_t) \qquad (2)$$

where W_y is a weight matrix and $\sigma(z)$ is the softmax function:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (3)$$

for $i = 1, \ldots, K$.

Because the model makes predictions about upcoming data, it is able to use the same data to generate and validate its predictions, allowing unsupervised learning. At every phoneme, the cross-entropy loss is assessed between the predicted distribution before encountering that phoneme and the phoneme's one-hot encoded identity y:

$$L(y, \hat{y}) = -y \cdot log(\hat{y}) \quad (4)$$

All models are trained in minibatches of 64 words, which are padded to have the same length as the longest word in the batch. Loss is aggregated across each batch and backpropagated to update W_x, W_h, W_y, and b_h. Models are optimized with Adam, a variant of stochastic gradient descent that maintains individual, adaptive learning rates for all parameters (Kingma and Ba, 2014).

We build and test two distinct types of models, both of which are variants of an RNNLM, differing in their representations of phonemes. In both cases, segment identities represented by one-hot vectors are mapped to columns of an embedding weight matrix W_E. These vectors serve as the inputs x_t for the computation in (1).

In *featural models*, the embedding vectors correspond to traditional ternary feature matrices, taken from the feature sets defined in Hayes (2009). We selected non-redundant subsets of these features for each language, and used them to construct a vector for each phoneme which specifies each feature value as positive (1), negative (-1), or underspecified (0). For example, the vector for English /b/ will have a 1 entry for the feature [VOICE], a -1 for [CONTINUANT], and a 0 for [HIGH], reflecting that [b] is a voiced non-continuant that is unspecified for height. These vectors are fixed during the learning process.

In *embedding models*, the columns of W_E can take on any value in $\mathbb{R}^e$, where e is a hyperparameter of the model. W_E is randomly initialized and optimized alongside other model parameters, following Bengio et al. (2003). This allows the models to learn segment representations from distributional information in a way that improves performance on the language modeling objective.

Embedding models have significantly more parameters than feature models. This makes direct comparison of the two classes of models difficult, and increases the risk that embedding models overfit. To mitigate this, and to produce more interpretable embeddings, we also report results from models where the input and output embeddings are tied, following Press and Wolf (2017). The embedding weight matrix W_E maps a one-hot vector of length n representing a phoneme's identity to a vector of length e. The output weight matrix W_y maps a hidden state vector h to a vector of length n, representing a distribution over phoneme identities. Tied embeddings require that $|h| = e$, which allows for shared weights such that $W_E = W_y^T$. This functions as a kind of regularization by restricting model parameters, forcing every mapping to and from the probability distribution over phonemes to use the same set of weights.

Hyperparameter settings were chosen to optimize performance while facilitating comparison across models. Embedding models of various sizes were evaluated on a randomized 60/40 training/development split of the English data. The model that assigned the highest likelihood to the development data had 24-dimensional embeddings and 64-dimensional hidden states. These parameters were used for all embedding models. For consistency, the featural models also have 64-dimensional hidden states. Tied embedding models are trained with 24-dimensional embeddings and hidden states, ensuring a similar number of parameters to featural models. For English, there are 9,320 parameters in the embedding model, 2,248 in the featural model, and 2,200 in the tied embedding model. The number of parameters in the featural model varies slightly between languages.

The featural and embedding models instantiate different predictions about the kinds of representations used in phonotactic grammars: the featural model assumes that subsegmental representations refer only to phonetic properties, while the embedding models allow these representations to be more abstract, conditioned on how each segment patterns in the observed data. Comparison of these models allows us to computationally investigate questions that are of theoretical interest to the field, such as to what extent different types of representation help or hinder the learning of phonotactic patterns (particularly those involving phonetically unnatural classes), and the importance of representations for generalization. We return to these points in the discussion in Section 7.

3 Evaluation data sets

We evaluate the models on three phonotactic data sets that exhibit phenomena that have proved challenging for previous models of phonotactics, or pose challenges for phonological theory more generally. These are Finnish vowel harmony (Section 4), Cochabamba Quechua laryngeal co-occurrence restrictions (Section 5), and English sonority projection (Section 6). Previous work suggests that models trained based on type frequency better predict human behavior than those trained on token frequency (Bybee, 1995; Albright and Hayes, 2003; Jarosz et al., 2017). We therefore do not take lexical frequency into account.

We compare the neural models against the Hayes and Wilson phonotactic learner (henceforth H&W; Hayes and Wilson, 2008). H&W is a commonly employed baseline in studies of phonotactic learning, and its use here allows the present work to be situated with respect to these studies (e.g., Albright, 2009; Daland et al., 2011; Futrell et al., 2017; Jarosz and Rysling, 2017).

H&W learns a set of featural constraints and associated weights from a training data set, and combines these constraints using a maximum entropy framework to assign probabilities to sequences of phonemes. We restrict constraint definitions to bigram or trigram windows. The Finnish and Cochabamba Quechua models learned 400 constraints, while the English model learned 600. H&W allows the analyst to specify tiers of segments over which constraints may be learned, facilitating the identification of long-distance phonotactic patterns. We compare results with and without a vowel tier for Finnish, and do not employ tiers for the other data sets.

Following Hayes and Wilson (2008), word scores for H&W are reported as maxent values (P^*), which for a word x is calculated as

$$P^*(x) = \exp\Big(-\sum_{i=1}^{N} w_i C_i(x)\Big) \quad (5)$$

where N is the number of constraints, w_i is the weight of the ith constraint, and $C_i(x)$ is the number of times word x violates the ith constraint. Maxent values are proportional to probabilities: higher values indicate higher probabilities.

The RNNLM word scores are reported as perplexity (ρ), which is the exponentiated entropy, or inverse of the mean log likelihood, of all phonemes in the test word.

$$\rho(x) = \exp\Big(-\sum_{i=1}^{|x|} \frac{1}{|x|} log_2(p(x_i))\Big) \quad (6)$$

Harmonic	Disharmonic
lumo	tumæ
hærø	mæntu
mekkottastu	vastekipæ
pømønøritæ	testurovevy

Table 1: Examples of harmonic and disharmonic Finnish nonce words in IPA.

Lower perplexities indicate higher probabilities.

The process of training H&W and the sRNN models is non-deterministic. H&W uses random sampling in the learning process, while the sRNN models have randomly initialized weights. We therefore report the mean scores from training and testing each model 10 times on each data set.

The model implementation and data sets are freely available online for use in future research.[2]

4 Finnish

4.1 Background

The first language we examine is Finnish. Finnish famously exhibits vowel backness harmony (e.g., Kiparsky, 1973; Ringen and Heinämäki, 1997; Goldsmith and Riggle, 2012). The language contains three classes of vowels: the front vowels {y, ø, æ}, the back vowels {u, o, a}, and the transparent vowels {i, e}. We refer to the set of front and back vowels as the harmonizing vowels. The vowels in a word generally agree in backness: that is, a word contains only transparent vowels and either front or back vowels. This restriction manifests in both root forms and affixing morphology.

This pattern is of interest because it is a long-distance phonotactic restriction. Not only can a number of consonants intervene between vowels, but an arbitrary number of transparent vowels may intervene between harmonizing vowels. This poses problems for n-gram models, which may not be able to detect illicit vowel subsequences if they are too far apart. We predict that the neural models will be better able to distinguish harmonic from disharmonic forms, particularly when sequences of transparent vowels occur.

4.2 Data

There is no publicly available corpus of transcribed Finnish. Because Finnish orthography is very close to a phonemic transcription, we instead

[2] https://github.com/MaxAndrewNelson/Phonotactic_LM

	Harm.	Disharm.	d
H&W tier (P^*)	0.00179	0.00105	0.46
H&W no tier (P^*)	0.802	0.708	0.23
Feat (ρ)	12.32	18.04	0.87
Emb (ρ)	14.97	25.93	0.86
Tied Emb (ρ)	11.03	14.42	0.79

Table 2: Average scores assigned by the models for Finnish harmonic and disharmonic words, along with effect size (Cohen's d).

use as training data a word list published by the Institute for the Languages of Finland.[3] We removed 584 words containing marginally attested characters, leaving 93,821 words in the corpus.

To test the models, we generated 20,000 nonce words, 10,000 harmonic and 10,000 disharmonic, ranging in length from 2–5 vowels (Table 1). Both sets are balanced for length. To ensure our models based their scores primarily on the harmony of words, we excluded CV sequences that were described to be impossible by a Finnish grammar (Suomi et al., 2008), and also excluded several CV sequences that were marginally attested in the corpus.[4] Syllables were either CV or CVC, with CC clusters drawn from the most common sequences in the corpus: /st/, /nt/, /tt/, and /kk/.

Because the test data is artificially generated, we perform no significance tests on these results. The size of the test set is arbitrary and consequently the power of the tests can be arbitrarily manipulated. Instead, we report effect sizes in the form of Cohen's d, which is the difference in group means expressed in units of pooled standard deviation (Cohen, 1988).

4.3 Results

The results are shown in Table 2. All models assign lower probabilities (lower maxent values and higher perplexities) to disharmonic forms. Cohen's d indicates that the RNNLMs make this distinction more robustly: by the heuristics in Cohen (1988), the featural and embedding models display a large effect size between harmonic and disharmonic scores ($d \geq 0.8$), and the tied model displays a medium effect size ($d \geq 0.5$), while the H&W models display a small effect size ($d \geq 0.2$). Allowing H&W to use a vowel tier produces a greater distinction between harmonic and disharmonic forms, though it substantially lowers the average maxent values assigned in the test corpus.

[3] http://kaino.kotus.fi/sanat/nykysuomi/

[4] These sequences are /fy/, /jø/, /fø/, /gø/, /fæ/, /gy/, /dø/, /gæ/, /bæ/, /by/, and /vø/.

	Span	Harm.	Disharm.	d
H&W (P^*) tier	1	0.00145	0.00131	0.12
	2	0.00138	0.00133	0.05
	3	0.00176	0.00196	0.16
H&W (P^*) no tier	1	0.746	0.707	0.09
	2	0.741	0.706	0.08
	3	0.804	0.758	0.13
Feat (ρ)	1	12.58	16.71	0.64
	2	13.10	16.31	0.38
	3	14.15	15.59	0.11
Emb (ρ)	1	15.79	21.21	0.57
	2	17.00	19.05	0.33
	3	16.47	18.94	0.20
Tied Emb (ρ)	1	11.49	13.42	0.61
	2	11.77	12.69	0.39
	3	11.75	12.61	0.36

Table 3: Model results for Finnish separated by the longest span of transparent vowels that intervene between two harmonizing vowels.

Table 3 shows that the models exhibit different performance on forms where harmonizing vowels are separated by one (e.g., [nøgihæ]; $n = 4189$), two (e.g., [jæsemehøpø]; $n = 644$), or three (e.g., [hydekistitø]; $n = 91$) transparent vowels. All models assign worse scores on average to disharmonic words, with the exception of the H&W tiered model, which assigns slightly higher scores to disharmonic words that contain spans of three transparent vowels. In addition, all models differentiate between harmonic and disharmonic forms less robustly as the maximum span of transparent vowels increases. In general, however, the RNNLMs are better able to differentiate between harmonic and disharmonic forms containing transparent vowels: the effect sizes for both H&W models on all spans is negligible ($d < 0.2$), while it is medium for all RNNLMs on spans of 1, and small on spans of 2 and 3. The exception is the featural model on spans of 3, which makes a negligible distinction. This suggests that the RNNLMs are better able to capture long distance dependencies than n-gram based models like H&W, even without the stipulation of a vowel tier.

5 Cochabamba Quechua

5.1 Background

The second language we examine is Cochabamba Quechua (CQ).[5] CQ has three series of stops (plain voiceless, aspirate, and ejective) at five places of articulation (labial, dental, postalveolar,

[5] Thanks to Gillian Gallagher for this data.

initial	medial	prohibited
t'anta	ɾit'i	*tant'a
k'atʃa	satʃ'a	*katʃ'a
pʰawaj	mosqʰoj	*posqʰoj
qʰari	ʎimpʰi	*tʃimpʰi

Table 4: Legal and prohibited laryngeal co-occurrence patterns in Cochabamba Quechua (Gallagher, 2019).

	Licit	Illicit (k)	Illicit (ʁ)
H&W (P^*)	0.67	0.28	0.30
Feat (ρ)	4.91	8.45	7.42
Emb (ρ)	4.89	8.45	7.55
Tied Emb (ρ)	4.91	8.28	7.16

Table 5: Model results for Cochabamba Quechua

velar, and uvular). These series participate in a laryngeal co-occurrence restriction in root forms: ejective and aspirated stops may occur either root-initially or root-medially, but they must be the first stop in the root (Table 4). Plain stops can occur following any type of stop (Gallagher, 2019).

The plain uvular stop in CQ is not realized as [q], but rather as [ʁ], a voiced uvular continuant. Gallagher (2019) provides phonetic, experimental, and phonological evidence that this phonetically disparate class (the plain stops plus [ʁ]) is active in speakers' synchronic grammars. CQ speakers preferred licit forms that do not violate the above laryngeal co-occurrence restriction to illicit forms that do, and they do not distinguish between k-initial and ʁ-initial illicit forms. For example, *[kap'a] and *[ʁap'a] are both judged as ill-formed by speakers, despite the latter appearing to satisfy the laryngeal co-occurrence restriction. Thus [ʁ] appears to pattern as a plain stop, despite being phonetically voiced and continuant.

This pattern is of interest because the set of plain stops that block the occurrence of subsequent aspirates and ejectives is a phonetically disparate class that cannot be captured with a conventional feature system, assuming [ʁ] is specified with features that reflect its phonetic realization. That is, the set of plain stops can only be specified by using disjunction between sets of features. This is primarily because [ʁ] is [+continuant], while the remaining plain stops are [–continuant]. We predict that the phonotactic models that use phonetic features may exhibit poorer performance on this pattern: specifically, we expect ʁ-initial illicit forms to receive better scores than k-initial illicit forms.

5.2 Data

We trained H&W and our three RNNLMs on a data set consisting of 2,468 CQ root forms. The data included two allophonic patterns related to uvular sounds: the vowels /i/ and /u/ surface as [e] and [o] respectively when adjacent to uvulars, and the sonorants /ʎ/, /w/, /j/, and /ɾ/ surface in uvularized forms before uvular sounds. These allophones were replaced by phonemic representations. This was done for the sake of allowing a smaller set of input segments and features to H&W, which scales poorly as the number of possible featurally-defined classes increases. This sanitization does not bear on the laryngeal co-occurrence pattern we are interested in. In addition, H&W recommends training on at least 3,000 input forms: we listed the frequency of each root as 2 in the input corpus to achieve this.

The trained models were tested on a set of 75 licit and illicit forms from Experiment 2 in Gallagher (2019). These forms were broken down into three classes: licit forms (e.g., [wap'a] or [pasi]), [k]-initial illicit forms (e.g., *[kap'a]), and [ʁ]-initial illicit forms (e.g., *[ʁap'a]). To determine whether the models assign significantly different scores to licit forms and the two types of illicit forms, we ran Kruskall-Wallis tests on each of the models with scores as the dependent variable and legality (licit vs. k-initial illicit vs. ʁ-initial illicit) as the independent variable. Kruskall-Wallis tests, which are the non-parametric equivalent of ANOVAs, were used because the scores violated several of the assumptions made by ANOVAs, such as normality of residuals. Post-hoc Dunn tests with Bonferroni correction were performed to identify significant pairwise differences.

5.3 Results

The results are shown in Table 5. Legality has a significant effect on score for all models (H&W: $\chi^2 = 14.53$, $p < 0.001$; Feat: $\chi^2 = 52.90$, $p < 0.001$; Emb: $\chi^2 = 53.17$, $p < 0.001$; Tied: $\chi^2 = 52.57$, $p < 0.001$). The H&W learner successfully distinguishes between licit and k-initial ($p < 0.01$) and ʁ-initial ($p < 0.05$) illicit forms, and does not make a distinction between k-initial and ʁ-initial illicit forms ($p > 0.05$). Similarly, all of the neural models are able to distinguish between licit and k-initial illicit forms (all models:

$p < 0.001$) and licit and ʁ-initial illicit forms (all models: $p < 0.001$), and not distinguish between k-initial and ʁ-initial illicit forms (all models: $p > 0.05$). Contrary to our prediction, laryngeal co-occurrence restrictions in CQ are learned by all models tested, even though this pattern makes reference to a phonetically disparate class. We can examine the models in more detail to gain insight into how this pattern is encoded in each case.

H&W cannot learn constraints that treat the plain stop series as a single class, because it cannot be uniquely specified by a feature matrix. The similar treatment of k-initial and ʁ-initial illicit forms results from multiple constraints that target different subsets of the plain stop series. For example, H&W consistently learned two high ranking constraints: *[–son, –cont]V[+CG], which penalizes illicit forms of a particular shape, except those with initial [ʁ]; and *[+dorsal, –syl]V[+CG], which penalizes only k-initial and ʁ-initial illicit forms of this shape (as well as legal but unattested forms like [xap'a]).

We may gain some insight into the neural models by comparing phoneme representations within each model using cosine similarity. Cosine similarity is the cosine of the angle between a pair of vectors: it is 1 when the vectors point in the same direction, 0 when they are orthogonal, and -1 when they point in opposite directions. We compare the embedding of [ʁ] with the mean of the embeddings of the classes of continuant and non-continuant consonants, which provide a representation of a 'typical' member of each class.

Table 6 shows that the representations of [ʁ] in the embedding models are more similar to the non-continuant consonants, while in the featural model it is more similar to the continuant consonants. We return to this point in the discussion.

	continuant	non-continuant
Featural [ʁ]	0.62	0.51
Emb [ʁ]	–0.20	0.31
Tied Emb [ʁ]	–0.26	0.19

Table 6: Cosine similarities between the embedding of [ʁ] and the mean embedding of the classes of continuant and non-continuant consonants in CQ. Learned embeddings are taken from individual runs of the models.

6 English

6.1 Background

The final phenomenon used to evaluate the neural models is English sonority projection. There is a strong preference cross-linguistically for syllables to have a sonority profile which increases monotonically from the left edge to the nucleus and then decreases from the nucleus to the right edge. This is known as the Sonority Sequencing Principle (SSP; Selkirk, 1984).

Effects of the SSP have been observed in acceptability judgments of novel words containing unattested clusters in Korean (Berent et al., 2008), Mandarin (Ren et al., 2010), English (Albright, 2007; Daland et al., 2011), and Polish (Jarosz and Rysling, 2017). The apparent universality of these effects and the fact that they apply to unattested clusters have led to a debate over whether these observations should be accounted for by an innate bias towards SSP conforming clusters (Berent et al., 2007, 2008), lexical statistics (Daland et al., 2011), or a combination of the two (Jarosz and Rysling, 2017).

We test our models on this case for two reasons. First, sonority sequencing is widely studied, particularly in English. This allows us to draw upon well-established experimental and modeling work to evaluate our results. Second, Daland et al. showed that the models that are best able to predict sonority projection from lexical statistics must have access to syllable structure and some form of subsegmental representation (for them, phonological features). Comparison of our featural and embedding models will allow us to test whether these representations must be based on phonetic properties, or if they may be learned statistically.

6.2 Data

All models were trained on 133,852 phonemically transcribed words in the Carnegie Mellon University Pronouncing Dictionary (CMU: Weide, 1998). Stress assignment information was removed. Words were not syllabified.

Trained models were evaluated against publicly available experimental results from Daland et al. (2011). These results come from an experiment designed to test the extent to which the sonority profile of onset clusters affects speaker acceptability judgements. Participants were tasked with choosing between pairs of nonsense words which each consisted of attested, unattested, and marginally attested English onset clusters of varying sonority profiles paired with one of six phonotactically licit tails. The onset clusters and tails tested are shown in Table 7. The total set of words contains 96 forms: each of the 48 onsets paired with two of the tails. For each word, Daland et al. (2011) derive an aggregate goodness score. This score reflects the proportion of trials in which a word containing that cluster was chosen over its competitor.

Onsets			**Tails**
Attested	Marginal	Unattested	
tw tr sw	gw ʃl	pw zr mr	-ɑtɪf
ʃr pr pl	vw ʃw	tl dn km	-ibɪd
kw kr kl	ʃn ʃm	fn ml nl	-ɑsɪp
gr gl fr	vl bw	dg pk lm	-ɛpɪd
fl dr br	dw fw	ln rl lt	-igɪf
bl sn sm	vr θw	rn rd rg	-ɛzɪg

Table 7: Stimuli from Daland et al. (2011).

6.3 Results

Trained models were used to score the stimuli in Table 7. The success of a model was determined by the linear correlation between the mean of the model's scores across runs and the goodness scores derived from human judgements. Table 8 reports the correlation coefficients (Pearson's r). Following Daland et al. (2011), we report separate coefficients for words containing attested, unattested, and marginal onset clusters, as well as global correlation coefficients. The maxent values produced by H&W are positively correlated with probability, while the perplexities produced by the neural models are inversely proportional to probability. We therefore present correlations as absolute values for the sake of readability.

	Overall	Attested	Unattested	Marginal
H&W (H)	0.759	0.000	0.686	0.362
Feat	0.868	0.354	0.823	0.551
Emb	0.866	0.365	0.765	0.609
Tied Emb	0.853	0.491	0.738	0.664

Table 8: Correlation coefficients between model and human ratings of novel words containing attested, unattested, or marginally attested complex onsets.

All of the neural models correlate better with human judgements than H&W on every partition of the data. The high correlations between neural and human judgements across all partitions of the data demonstrate that subsegmental representations based on the phonetic properties of sounds are not necessary to effectively learn the SSP: suitable embeddings can also be learned solely from lexical statistics. This is in agreement with the findings of Mirea and Bicknell (2019), although they do not partition the data by onset type.

This is not to say, however, that there are no differences in performance between prespecified and learned embeddings. There is a tendency for the embedding models to fit observed clusters better (the attested and marginal partitions), while the featural model appears to generalize to unattested forms more effectively.

Because the available data from Daland et al. (2011) is aggregated, we are unable to use bootstrap methods to estimate the ceiling correlation coefficient, which would shed light on the extent to which human judgements would be expected to correlate with other human judgements.

	Overall	Attested	Unattested	Marginal
H&W	0.83	0.000	0.76	0.02

Table 9: Correlation coefficients between model and human judgements from the best performing model in Daland et al. (2011).

Neural models not only outperform our implementation of H&W, but perform comparably to Daland et al.'s best reported model result (Table 9), which used a version of H&W that was supplied with syllable structure. Overall these results suggest that neural phonotactic language models are able to predict aggregate human behavior as well or better than existing models even when provided with less structured input data, and that this performance does not crucially depend on whether subsegmental representations correspond to phonetic properties.

7 Discussion and conclusion

RNN language models can learn and generalize phonotactic patterns as well as or better than H&W across all cases considered here. The use of RNNs is particularly beneficial in the cases of Finnish and English. In Finnish, the ability of the RNN models to represent long distance dependencies allowed them to better generalize the harmony pattern to novel forms. In English, H&W generally assigns perfect scores to attested and (to a lesser extent) marginal forms, while the RNNLMs assign scores which better correlate with human judgements. Although prediction of human judgements is not the only goal of phonotactic modeling, it is an important one, and we believe these are useful improvements.

Comparing the performance of the models tested in this paper also provides predictions relevant to theories of universal vs. language-specific features (e.g., Mielke, 2008; Archangeli and Pulleyblank, 2018; Mayer and Daland, in press), and how this relates to the division of phonological labor between constraints and representations. The general success of the embedding models across tasks suggests these patterns may be effectively learned with no reference to segments' phonetic properties. However, it is also true that the models where segments were represented in terms of their phonetic properties were able to learn patterns involving a phonetically disparate class. The existence of such classes is a central motivation for theories of learned features.

H&W captures the CQ pattern by learning a set of constraints that, acting in tandem, produce the correct pattern. This is reminiscent of the phonological conspiracies raised by Kisseberth (1970), in that the homogeneous behavior of the plain stop series (including [ʁ]) emerges from the interaction of a set of apparently independent constraints, rather than a unified treatment by the grammar. The featural RNNLM also lacks a unified representation of this class, and we may assume the homogeneous behavior is generated by the processes applied to the representations (though these processes are computationally different from H&W). The embedding models, on the other hand, shift some of the work onto the representations, learning embeddings for [ʁ] that reflect distributional rather than phonetic properties.

Thus these models characterize different hypotheses about how phonetically disparate classes are distributed between representations and processes (e.g., rules or constraints) in the grammar. Although the performance of the featural and embedding models is indistinguishable for CQ, the results from English suggest that phonetic features may allow the models to generalize more effectively, at the expense of a poorer fit to observed data (see, e.g., Mitchell, 1980). We are optimistic that further modeling (perhaps combining fixed and learned embeddings) and comparison with human judgements will provide additional insight.

Another contribution of this paper is to show that sRNNs are able to learn phonotactic patterns as effectively as more complex models such as LSTMs (cf. Mirea and Bicknell, 2019). Phonotactic patterns are generally less complex than the syntactic/semantic patterns central to language modeling research (Heinz and Idsardi, 2013), and sRNNs may provide an appropriate fit to this complexity. For example, Weiss et al. (2018) demonstrate that, unlike LSTMs, sRNNs are unable to learn the $a^n b^n$ pattern, which is known to be phonotactically unattested (Eisner, 1997; Lamont, 2019). We anticipate for this reason that the use of more advanced models, such as attention-based language models (Vaswani et al., 2017), will not necessarily entail better performance on phonotactic learning and generalization.

Much work remains to be done. A concern with RNNLMs is that they are not as transparent as models like H&W, and are therefore of less theoretical value. Developing methods to gain insight into what these models have learned, such as probe or clustering tasks, is an important next step for their application to phonotactic learning. Such tasks can negate the interpretability problems associated with neural networks and allow access to what linguistic information is being encoded (e.g., Alishahi et al., 2019; Nelson and Mayer, 2019).

In particular, we have only shown that these models match human-like behavior in aggregate. It will be useful to explore how they deviate from human behavior in specific cases. We also note that the neural models we present here operate from left-to-right, and may have difficulty with regressive phonotactic patterns. Bidirectional RNNs (Schuster and Paliwal, 1997) have the potential to overcome this limitation.

The power of neural models as statistical learners provides a valuable tool for work on the learnability of linguistic phenomena by allowing us to begin determining the upper limit on what is learnable from lexical statistics alone, and how different representational assumptions guide this learning. We share Pater (2019)'s enthusiasm for the ongoing integration of neural research with linguistic theory as a supplement to more traditional methodology.

Acknowledgements

We thank Gillian Gallagher, Bruce Hayes, Gaja Jarosz, Joe Pater, and the attendees of the UMass Sound Workshop. We also thank three anonymous reviewers for their valuable feedback and criticism. The authors are listed in alphabetical order.

References

Adam Albright. 2007. Natural classes are not enough: Biased generalization in novel onset clusters. In 15th Manchester Phonology Meeting, Manchester, UK, pages 24–26.

Adam Albright. 2009. Feature-based generalization as a source of gradient acceptability. Phonology, 26:9–41.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. Cognition, 90:119–161.

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. arXiv preprint arXiv:1904.04063.

Diana Archangeli and Douglas Pulleyblank. 2018. Phonology as an emergent system. In S.J. Hannahs and Anna R.K. Bosch, editors, The Routledge Handbook of Phonological Theory, pages 476–503. Routledge, London.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137–1155.

Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166.

Iris Berent, Tracy Lennertz, Jongho Jun, Miguel A. Moreno, and Paul Smolensky. 2008. Language universals in human brains. Proceedings of the National Academy of Sciences, 105:5321–5325.

Iris Berent, Donca Steriade, Tracy Lennertz, and Vered Vaknin. 2007. What we know about what we have never heard: Evidence from perceptual illusions. Cognition, 104:591–630.

Joan Bybee. 1995. Regular morphology and the lexicon. Language and Cognitive Processes, 10:425–455.

Noam Chomsky and Morris Halle. 1965. Some controversial questions in phonological theory. Journal of Linguistics, 1:97–138.

Jacob Cohen. 1988. Statistical power analysis for the behavioral sciences, 2nd edition. Erlbaum, Hillsdale, NJ.

John Coleman and Janet Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In John Coleman, editor, Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology, pages 49–56. Association for Computational Linguistics, Somerset, NJ.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. Phonology, 28:197–234.

Jason Eisner. 1997. What constraints should OT allow? Handout (20p) for talk at the LSA Annual Meeting, Chicago, 1/4/97. (ROA-204-0797).

Jeffrey L. Elman. 1990. Finding structure in time. Cognitive Science, 14(2):179–211.

Richard Futrell, Adam Albright, Peter Graff, and Timothy J. O'Donnell. 2017. A generative model of phonotactics. Transactions of the Association for Computational Linguistics, 5:73–86.

Gillian Gallagher. 2019. Phonotactic knowledge and phonetically unnatural classes: the plain uvular in Cochabamba Quechua. Phonology, 36:37–60.

John Goldsmith and Jason Riggle. 2012. Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. Natural Language and Linguistic Theory, 30:859–896.

Bruce Hayes. 2009. Introductory Phonology. Wiley-Blackwell, Malden, MA.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. Linguistic Inquiry, 39(3):379–440.

Jeffrey Heinz and William Idsardi. 2013. What complexity differences reveal about domains in language. Topics in Cognitive Science, 5(1):111–131.

Gaja Jarosz, Shira Calamaro, and Jason Zentz. 2017. Input frequency and the acquisition of syllable structure in Polish. Language Acquisition, 24:361–399.

Gaja Jarosz and Amanda Rysling. 2017. Sonority sequencing in Polish: the combined roles of prior bias and experience. In Karen Jesney, Charlie O'Hara, Caitlin Smith, and Rachel Walker, editors, Supplemental Proceedings of the 2016 Annual Meeting on Phonology. Linguistic Society of America, Washington, DC.

Frederick Jelinek. 1999. Statistical methods for speech recognition. MIT Press, Cambridge, MA.

Dan Jurafsky and James Martin. 2008. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech processing, 2nd edition. Prentice-Hall, Upper Saddle River, NJ.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Paul Kiparsky. 1973. Phonological representations. In Osamu Fujimura, editor, Three Dimensions of Linguistic Theory, pages 1–136. TEC, Tokyo.

Charles W. Kisseberth. 1970. On the functional unity of phonological rules. Linguistic Inquiry, 1(3):291–306.

Andrew Lamont. 2019. Majority rule in harmonic serialism. In Katherine Hout, Anna Mai, Adam McCollum, Sharon Rose, and Matthew Zaslansky, editors, Supplemental Proceedings of the 2018 Annual Meeting on Phonology. Linguistic Society of America, Washington, DC.

Connor Mayer and Robert Daland. in press. A method for projecting features from observed sets of phonological classes. Linguistic Inquiry.

Jeff Mielke. 2008. The emergence of distinctive features. Oxford University Press, Oxford.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In Proceedings of INTERSPEECH, pages 1045–1048.

Nicole Mirea and Klinton Bicknell. 2019. Using LSTMs to assess the obligatoriness of phonological distinctive features for phonotactic learning. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1595–1605.

Tom M. Mitchell. 1980. The need for biases in learning generalizations. Report C BM-TR-5-110. New Brunswick, NJ: Rutgers University, Department of Computer Science.

Max Nelson and Connor Mayer. 2019. Learning and generalizing phonotactics with recurrent neural networks. Poster presented at the 2019 Annual Meeting on Phonology. Stonybrook, NY.

Joe Pater. 2019. Generative linguistics and neural networks at 60: foundation, friction, and fusion. Language, 93:41–74.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Lawrence Phillips and Lisa Pearl. 2015. The utility of cognitive plausibility in language acquisition modeling: Evidence from word segmentation. Cognitive Science, 39:1824–1854.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Jie Ren, Liqun Gao, and James L. Morgan. 2010. Mandarin speakers' knowledge of the sonority sequencing principle. In 20th Colloquium on Generative Grammar.

Catherine O. Ringen and Orvokki Heinämäki. 1997. Variation in Finnish vowel harmony: An OT account. Natural Language & Linguistic Theory, 17:303–337.

Natalie M. Schrimpf and Gaja Jarosz. 2014. Comparing models of phonotactics for word segmentation. In Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM, pages 19–28. Association for Computational Linguistics, Baltimore.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45:2673–2681.

Elisabeth Selkirk. 1984. On the major class features and syllable theory. In Mark Aronoff and Richard T. Oehrle, editors, Language sound structure: Studies in phonology presented to Morris Halle by his teacher and students, pages 107–113. MIT press, Cambridge, MA.

Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In Proceedings of the Society for Computation in Linguistics (SCiL) 2018.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association.

Kari Suomi, Juhani Toivanen, and Riikka Ylitalo. 2008. Finnish sound structure. Studia humaniora ouluensia, 9.

Daniel Swingley. 2005. Statistical clustering and the contents of the infant vocabulary. Cognitive Psychology, 50:86–132.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Michael S. Vitevitch and Paul A. Luce. 2004. A Web-based interface to calculate phonotactic probability for words and nonwords in English. Behavior Research Methods, Instruments, and Computers, 36:481–487.

Robert L. Weide. 1998. The CMU pronouncing dictionary. http://www.speech.cs.cmu.edu/cgibin/cmudict.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Volume 2, Short Papers, pages 740–745.

Charles D. Yang. 2004. Universal Grammar, statistics, or both. Trends in Cognitive Sciences, 8(10):451–456.

An IBSP Description of Sanskrit /n/-Retroflexion

Ayla Karakaş
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
ayla.karakas@stonybrook.edu

Abstract

Graf and Mayer (2018) analyze the process of Sanskrit /n/-retroflexion (*nati*) from a subregular perspective. They show that *nati*, which might be the most complex phenomenon in segmental phonology, belongs to the class of *input-output tier-based strictly local languages* (IO-TSL). However, the generative capacity and linguistic relevance of IO-TSL is still largely unclear compared to other recent classes like the *interval-based strictly piecewise* languages (IBSP: Graf, 2017, 2018). This paper shows that IBSP has a much harder time capturing *nati* than IO-TSL, due to two major shortcomings: namely, the requirement of an upper bound on relevant segments, and a lack of descriptive succinctness.

1 Introduction

Research in computational phonology has determined that all phonological patterns fit in the class of finite-state languages (Kaplan and Kay, 1994). The study of subregular phonology explores tighter characterizations of phonological phenomena in the form of subclasses of the regular languages. This furnishes lower and upper complexity bounds for phonological computations, which in turn provides new insights for typology and learnability — see Heinz 2018 and references therein.

One phenomenon that has proven to be particularly complex is /n/-retroflexion in Sanskrit, also known as *nati*. The nasal /n/ undergoes retroflexion whenever it appears immediately before a sonorant and a retroflex exists somewhere to its left. While this interaction of local and non-local factors is already unusual, the true complexity of the process comes from various blocking effects. It has been known since Graf (2010) that *nati* — when viewed as a phonotactic constraint on surface forms — is star-free. Recently, an alternative upper bound has been established in the form of *input-output tier-based strictly local languages* (IO-TSL; Graf and Mayer, 2018).

IO-TSL is an extension of the empirically well-supported class TSL (Heinz et al., 2011). Whereas subclasses of IO-TSL enjoy independent empirical support (De Santo and Graf, 2019; Mayer and Major, 2018), the only empirical motivation for IO-TSL itself is *nati*. The formal properties of IO-TSL are also not well-understood. It is not even known whether IO-TSL is a subclass of the star-free languages. By contrast, the class of *interval-based strictly piecewise* languages (IBSP; Graf, 2017, 2018) is properly star-free, handles a wide range of phonotactic phenomena, and has even been applied to syntax (Shafiei and Graf, 2019). For all these reasons, an IBSP analysis of *nati* would be a valuable addition to the current IO-TSL description, and might furthermore shed light on how these two classes differ.

In this paper, I argue that *nati* belongs to the intersection closure of IBSP, but the resulting grammar is much more convoluted than the IO-TSL analysis. While the basic cases of *nati* are very natural from an IBSP perspective, the interactions of blocking effects are hard to capture due to two limitations of IBSP's notion of *open slots*: the inability to force a segment to always appear in an open slot, and the inability to mark an open slot as optional. These insights might prove useful for a future proof separating IBSP and IO-TSL.

The structure of the paper is as follows: IBSP is formally defined in Sec. 2, adapting the more general format proposed in Graf (2018). Sec. 3 then walks the reader through the *nati* analysis, starting from the simplest case and refining the IBSP grammar with each new complication. Sec. 4 reflects on the status of the analysis and what limitations of IBSP make *nati* so difficult to account for.

2 Preliminaries

Graf (2017) first defined the class of *interval-based strictly piecewise* (IBSP) string languages as an extension of the *strictly piecewise* (SP) languages (Rogers et al., 2010). IBSP enriches SP with locality domains, and the checking of SP-dependencies is limited to these locality domains. IBSP properly subsumes SP, but also the classes SL and TSL, all three of which play a major role in subregular phonology. Graf (2018) further generalizes the format of locality domains to account for phenomena that had previously been analyzed in terms of I-TSL. Only this more general version can handle *nati*.

Intuitively, an IBSP interval involves definitions of I) the left and right *domain edge*, II) a finite number k of *open slots*, and III) the *fillers* that can occur between open slots. Fillers and domain edges are defined through *k-intervals*, also called k-vals. The IBSP grammar also supplies a list of forbidden k-grams. A string is well-formed iff there is no way to instantiate the k-val in such a manner that the configuration of open slots matches a forbidden k-gram.

While IBSP is originally defined in terms of first-order logic (Graf, 2017), I adopt the newer definition of Shafiei and Graf (2019) as it also subsumes the generalized intervals of Graf (2018). Note that $\cdot$ in definition 2.2 denotes string concatenation lifted to sets, i.e. $S \cdot T := \{st \mid s \in S, t \in T\}$.

Definition 2.1 (k-val)**.** A *segmented k-interval* ($k \geq 0$) over alphabet Σ, or simply segmented k-val, is a tuple $\langle L, R, F_i \rangle_{0 \leq i \leq k}$ such that:

- $L, R \subseteq \Sigma \cup \{\varepsilon\}$ specify the left edge and right edge, respectively, and
- $F_i \subseteq \Sigma$ specifies the i-th filler slot.

Definition 2.2 (IBSP-k)**.** Let Σ be some fixed alphabet and $\rtimes, \ltimes \notin \Sigma$ two distinguished symbols. An IBSP-k grammar over $\Sigma \cup \{\rtimes, \ltimes\}$ is a pair $G := \langle i, S \rangle$, where i is a segmented k-val over $\Sigma \cup \{\rtimes, \ltimes\}$ and $S \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a set of forbidden k-grams. A string $s \in \Sigma^*$ is generated by G iff there is no k-gram $u_1 ... u_k \in S$ such that $\rtimes^k s \ltimes^k$ is a member of the language

$$(\Sigma \cup \{\rtimes, \ltimes\})^* \cdot L \cdot F_0^* \cdot \{u_1\} \cdot F_1^* \cdot \{u_2\} \cdot \\ \ldots \cdot F_{k-1}^* \cdot \{u_k\} \cdot F_k^* \cdot R \cdot (\Sigma \cup \{\rtimes, \ltimes\})^*$$

The language $L(G)$ is the set of all $s \in \Sigma^*$ that are generated by G. A stringset L is IBSP-k iff $L = L(G)$ for some IBSP-k grammar G.

The reader may skip ahead to (1) and (2) for a depiction of a concrete IBSP interval and its application to an illicit string.

In IBSP, all possible instantiations of a locality domain must be evaluated. If at least one of them yields a match for an illicit k-gram, the whole string is discarded. By default, fillers allow each open slot to be arbitrarily far away from the next one. However, adjacency of the i-th and $i+1$-th open slot can be enforced by stipulating $F_{i+1} = \emptyset$. Here, F_{i+1} refers to the subset of Σ that is allowed in the filler between the i-th and $i+1$-th slots. The subset is empty if nothing is allowed in that filler. This is not to be confused with the string language corresponding to the $i+1$-th filler, which is $F_{i+1}^* = \{\varepsilon\}$. Mixing such empty fillers with normal fillers allows IBSP to capture phonotactic constraints in which local and non-local dependencies interact. As we will see next, this is not needed for the simplified version of *nati*, but will be crucial once the full range of facts is considered (Sec. 3.3 and subsequent sections).

3 Data and Analysis

Nati is a left-to-right long-distance assimilation process with a single trigger, a single target, and several conditions for blocking. While *nati* is usually described as a process — i.e. a mapping from underlying forms to surface forms — I treat it as a phonotactic phenomenon. That is to say, *nati* is reanalyzed as a constraint on the distribution of [n] in surface forms, making it a matter of string languages rather than string transductions. This is in line with the previous work done by Graf and Mayer (2018), which will henceforth be referred to as G&M.

The discussion starts with the simplest cases of *nati* and continually refines the IBSP description as new data is considered. The final version is presented in Sec. 3.5.

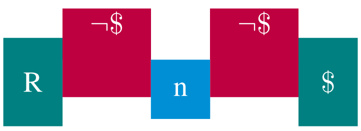
Several notational conventions will be adopted for the remainder of this paper: Sanskrit examples have their triggers and targets bolded, while active blockers are underlined. All the examples are taken from G&M and Ryan (2017). Since the phonotactic perspective forgoes any notion of underlying forms, I will only use square brackets to denote surface segments throughout this paper. IBSP interval diagrams are represented in a pictorial fashion: domain edges are large, green rectangles, fillers are vertically offset boxes in red, and open slots are blue squares.

3.1 Long-distance assimilation

Nati starts out with the basic constraint that a nasal target /n/ becomes [ɳ] when preceded arbitrarily far to the left by a non-lateral retroflex continuant in {/ɻ/, /ɻ̩/, /ɻ̩ː/, /ʂ/}. G&M formalize this as the constraint "no [n] may appear in the context $R \cdots \_$", where R is one of the triggers listed in the preceding sentence.

G&M's constraint is easily expressed in terms of IBSP. Our grammar consists of a single forbidden unigram, which is n. By keeping word edges ($) and string edges ({⋊, ⋉}) distinct, IBSP enables us to instantiate intervals across multiple words in a string, if desired. I will use $ instead of ⋉ for now as this does not commit us as to whether the string consists of a single phonological word or a sequence of words. But as discussed in Sec. 4, it may eventually be necessary to use the string edge ⋉ instead. For now, the use of the word edge $, along with banning the appearance of $ in fillers, captures that *nati* cannot apply across word boundaries.

(1) **IBSP interval (Version 1)**



For the sake of succinctness, the interval above lists the forbidden unigram directly in the open slot. While this is non-standard, I believe it makes the analysis easier to follow once the complexity of the intervals starts to increase.

Tab. 1 lists some data points that are relevant for this base case. The form of the instrumental singular suffix /-eːna/ alternates based on whether the root it attaches to contains a trigger for *nati*. For the sake of exposition, I also include an illicit nonce variation, indicated by the gloss "N/A".

Form	**Gloss**	***Nati*?**	**Licit?**
káːm-eː**n**a	'by desire'	✗	✓
manuʂj-eː**ɳ**a	'by human'	✓	✓
manuʂj-eː**n**a	N/A	✓	✗

Table 1: Forms showing basic *nati* (Ryan, 2017, p. 305)

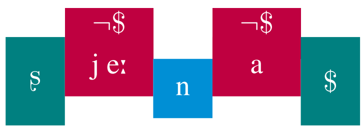
The reader may wonder why an analogous nonce form *káːm-eːɳa* is not included in Tab. 1. In this nonce form, /n/ would undergo *nati* without a suitable trigger, which should be illicit. However, this presupposes a view of *nati* as a process. From the perspective of phonotactics, it is not obvious that this nonce form is actually illicit because [ɳ] can occur independently of *nati*. The phonotactics of *nati* only concern the distribution of [n], not [ɳ], so only the former need to be considered here.

Let us now see how the locality domain in (1) captures the well-formedness of the first two forms in Tab. 1 while also ruling out the illicit nonce form. First, *káːm-eːna* is well-formed because it lacks a retroflex, so there is no suitable left edge for the interval in (1). Hence the locality domain cannot be established at all, so there are no open slot configurations to check against the list of forbidden unigrams. As a result, the string is well-formed.

The second example is *manuʂj-eːɳa*, which does allow for numerous instantiations of the interval. In all instantiations, the interval spans from [ʂ] to the right word edge, and the only difference is what segments make up the fillers and which one ends up in the open slot. Since *manuʂj-eːɳa* does not contain any [n], the open slot never matches the forbidden unigram. Consequently, this string is also deemed well-formed. In contrast to the first example, where well-formedness followed from the inability to instantiate any locality domain, this example allows for many distinct instantiations but none of them yield a forbidden configuration of open slots.

This leaves us with the illicit *manuʂj-eːna*. It works exactly like the second case, except that now there is an instantiation that results in a match with the forbidden unigram n. This particular instantiation is depicted below.

(2) **IBSP interval: manuʂj-eːna**



So far, IBSP has not done anything that could not be accomplished by simpler means, e.g. an SP grammar. As we start adding on conditions and exceptions, though, IBSP intervals will quickly become indispensable.

3.2 Unconditional blocking by intervening coronals

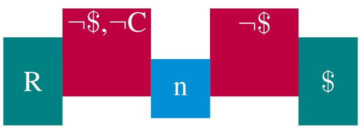
We now turn to the first of the *nati*-blocking effects: /n/-retroflexion can be blocked if a coronal segment appears between trigger and target. The set of relevant coronals includes retroflexes but excludes the glide [j] as the latter is both a sonorant and a coronal — see Ryan (2017) for further discussion. Tab. 2 lists a particular example of coronal blocking, an illicit nonce form, and a nonce form that illustrates what the surface form would look like if coronals were not blockers.

Form	Gloss	*Nati*?	Blocking?	Licit?
ʋaɻɳ-a**n**aːnam	no gloss	✗	✓	✓
ʋaɻm-a**n**aːnam	N/A	✗	✗	✗
ʋaɻɳ-a**ɳ**aːnam	N/A	✓	N/A	✓

Table 2: Forms showing blocking by intervening coronals (Hansson, 2001, p. 227)

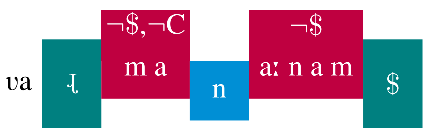
In G&M, the forbidden context for [n] is updated to $R\overline{C}\cdots\_$, where $\overline{C}$ matches every segment that is not a coronal, including [j]. To represent this in IBSP, we modify the first filler in (1) so that it may not contain any coronals either. If a string contains a coronal, it must go in the open slot or the second filler. Either way, no subsequent [n] can appear in the open slot, and consequently the string will be deemed well-formed.

(3) **IBSP interval (Version 2)**



At the same time, strings without coronals will still be judged illicit. This is illustrated below for the nonce form *ʋaɻm-anaːnam*.

(4) **IBSP interval: ʋaɻm-anaːnam**



Note that [ɳ] itself is a coronal blocker, so any subsequent [n] in a word loses its eligibility as a target for *nati*. The only exception to this is geminate [nn] sequences where both [n] become retroflexed. However, this could also be treated as a separate process of progressive local assimilation. I put this issue aside for now, but it will be revisited in Sec. 4.

3.3 Mandatory adjacency to sonorant

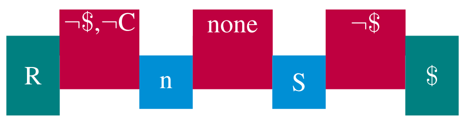
In order for [n] to undergo *nati*, it must also be immediately followed by a vowel, a glide, [m], or [n] itself. More succinctly, the following segment must be a non-liquid sonorant (Whitney, 1889). For example, in the form *bɻahman*, *nati* does not apply as [n] occurs at the very end of the word without any subsequent sonorant. Similarly, *nati* does not apply in *caɻ-a-n-ti*, in this case because [t] is not a sonorant. Sanskrit has some nasals besides [m] and [n] that are non-liquid sonorants, but since those cannot follow [n] for independent reasons (Emeneau, 1946) they do not matter for the purposes of this paper.

Form	Gloss	*Nati*?	Sonorant?	Licit?
caɻ-a-**n**-ti	'wander (3Pl)'	✗	✗	✓
bɻahma**n**_	'brahman'	✗	✗	✓
bɻahma**n**a	N/A	✗	✓	✗

Table 3: Forms showing mandatory adjacency to sonorant; (Hansson, 2001, p.229) and (Ryan, 2017, p. 318)

G&M represent the new illicit context for [n] as $R\overline{C}\cdots\_S$, where S is a suitable sonorant. We will use the same definition of S to add a second open slot to the interval in (3). The list of illicit unigrams is now expanded to illicit bigrams. It is no longer just [n] that is forbidden, but rather any bigram of the form nS. Keep in mind that coronal blocking is still active, though.

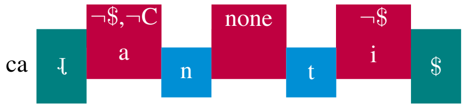
(5) **IBSP interval (Version 3)**



The descriptor *none* in the second filler of (5) indicates that $F_1 \subset \Sigma$ is $\emptyset$ (and thus $F_1^* = \{\varepsilon\}$). That is to say, this filler cannot contain any symbols at all and the first and second open slot must always be adjacent.

Let us verify that the first two examples in Tab. 3 are still well-formed given the grammar in (5). Below is an example of one possible interval established in *caɻ-a-n-ti*.

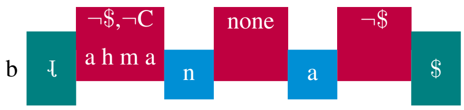
(6) **IBSP interval: caɻ-a-n-ṯi**



This is the only interval that could possibly cause the IBSP grammar to reject the string, since the first open slot is filled by n. However, as the second open slot is not a sonorant, the open slot configuration does not match any of the forbidden bigrams. The well-formedness of *bɻahman* follows for the very same reason: there is no way of instantiating the locality domain so that the two open slots would contain [n] and a sonorant, respectively.

At the same time, *bɻahmana* is correctly ruled out as illicit.

(7) **IBSP interval: bɻahmana**



3.4 Conditional blocking by preceding velar and labial plosives

Coronal consonants are not the only blockers of *nati*: velar and labial plosives also block its application, but only if I) the plosive immediately precedes the target nasal, and II) a left root boundary (√) occurs somewhere between the trigger and the plosive. Based on the data given in G&M and Ryan (2017), I assume that for a given word, an interval instantiated within the word never has to contend with more than one √ — this will be elaborated on in Sec. 4. Blocking is contingent on both conditions being met, as is exemplified by the data in Tab. 4. In *pɻa-√miːɳ-aː-ti*, *nati* still occurs across a left root boundary due to the absence of a plosive immediately before [n]. In *√ɻug-ɳá*, *nati* can target an n after an immediately preceding velar plosive [g] because the left root boundary does not occur between the triggering retroflex and the plosive. Only in *(abʱi-)pɻa-√gʱn-an-ti* does *nati* fail as there is both a plosive and a root boundary, both of which occur in the relevant positions.

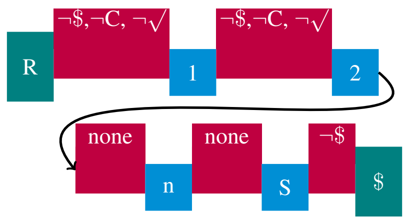
Form	Gloss	*Nati?*	Licit?
pɻa-√miːɳ-aː-ti	'vanishes (3s)'	✓	✓
√ɻug-ɳá	'break (pass. part.)'	✓	✓
(abʱi-)pɻa-√gʱn-an-ti	'broken'	✗	✓

Table 4: Forms showing conditional blocking by preceding plosives (Ryan, 2017, p. 319, 321)

In response to this additional complication, G&M update the banned context to $R\alpha \cdots \_$. Here α is any string that neither contains a coronal nor matches $\cdots\surd\cdots P$, with P denoting a velar or labial plosive. It is at this point that the complexity of our IBSP treatment ramps up significantly. We must now introduce open slots whose only purpose is to be sensitive to the conditional presence of certain segments. By setting up the fillers in such a way that root boundaries and immediately preceding plosives can only go into open slots, we can ensure that the grammar is always aware of these segments if they occur in the string. The list of forbidden k-grams is then set up in such a fashion that open slot configurations that start with a root boundary and a plosive are exempt from *nati*. This is a very unusual use of open slots and fillers, and I am unaware of any other IBSP-analysis that has to resort to this trick.

The concrete steps are as follows. First, two additional open slots must be included between the trigger and target. Open slot 1 detects the presence of a left root boundary somewhere arbitrarily to the left of [n]. Open slot 2 detects the presence of a velar/labial plosive immediately before an [n]. For readability, graphical depictions of longer intervals will now be broken up across two lines.

(8) **IBSP interval (Version 4)**



The filler before the third open slot is set to *none* so that it can only be filled by whatever segment immediately precedes [n]. The fillers surrounding the first open slot are more complex. The ban against coronals is carried over from coronal blocking, but in addition these fillers may not contain a root boundary either. As a result, a root boundary that occurs somewhere between the triggering retroflex and a suitable plosive is forced into the first open slot. The conjunction of all these factors ensures that if a string contains a suitable root boundary and plosive, they will always occur in the first two open slots.

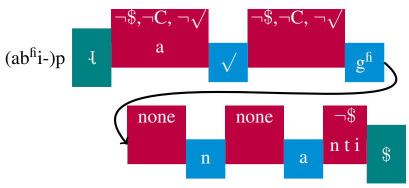
In the next step, we expand the list of forbidden bigrams of the form nS to forbidden 4-grams of the form ϕnS. Here ϕ represents a large number of bigrams. As *nati* is only blocked whenever the first open slot is a root boundary and the second open slot is a plosive, nS is illicit if:

1. the first open slot is not a root boundary, or
2. the second open slot is not a plosive, or
3. both 1 and 2 hold.

Hence ϕ corresponds to any combination of segments that matches one of the three conditions above.

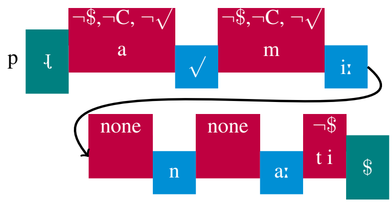
If the first two open slots in an instantiated interval do not match ϕ, *nati* will not be enforced, capturing the described blocking effect. This is illustrated below for *(abʱi-)pɻa-√gʱn-an-ti*.

(9) **IBSP interval: (abʱi-)pɻa-√gʱn-an-ti**



Any configuration where the first two open slots are not √ and a plosive will match ϕ, triggering a *nati* violation if the remaining two open slots are filled by [n] and a sonorant. As a concrete example, consider the nonce form *pɻa-√miːn-aː-ti*.

(10) **IBSP interval: pɻa-√miːn-aː-ti**



The reader is urged to verify for themselves that the remaining forms in Tab. 4 are handled correctly by this grammar.

An additional bug arises in that the introduction of new open slots has created an "escape hatch" for coronals. In previous versions, a coronal had to go into the first or second open slot, or the third filler. These are now the third and fourth open slot and the fifth filler. While coronals are still banned in the first and second filler, they could go into the first or second open slot. Since ϕ currently matches coronals, too, we no longer capture coronal blocking. Fortunately, the fix is easy. We further restrict the shape of ϕ so that it does not match any open slot configuration with a coronal. Overall, this leaves the following patterns for ϕ:

1	2
$\surd$	$\neg$P $\wedge$ $\neg$C
$\neg\surd \wedge \neg$C	P
$\neg\surd \wedge \neg$C	$\neg$P $\wedge$ $\neg$C

Figure 1: Open slots in ϕ s.t. nS is illicit

Given a list of suitable list of segments for Sanskrit, ϕ can be compiled out into a list of bigrams. These bigrams are then prefixed with every possible instantiation of nS to arrive the list of forbidden 4-grams.

3.5 Conditional blocking by following retroflex

Even though the grammar in (8) is already fairly complicated, it still does not handle the last layer of *nati*: if a retroflex appears arbitrarily far to the right of the target [n], /n/-retroflexion may be blocked. Blocking only occurs when both of the following two conditions are met: I) a left root boundary intervenes between the trigger and the target, and II) there is no coronal between the target [n] and blocking retroflex. Condition II) is particularly peculiar. Essentially, the appearance of a coronal consonant between [n] and its following retroflex blocks the blocking of *nati* by said retroflex, so that *nati* applies as usual.

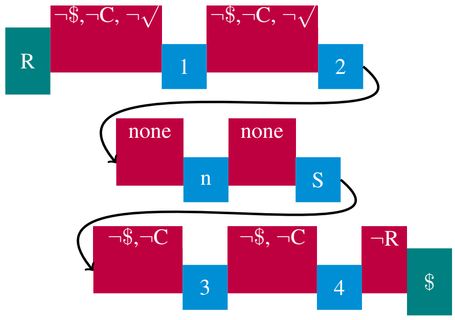
Form	Gloss	*Nati*?	Licit?
pɻa-√**n**aṣ-ṭum	'to vanish (inf.)'	✗	✓
pɻa-√**ɳ**eː-tɻ̩	'leader'	✓	✓
pɻ̩-ɳa-k-ṣi	'unite (2s)'	✓	✓

Table 5: Forms showing conditional blocking by following retroflex (Ryan, 2017, p. 325)

The form *pɻa-√naṣ-ṭum* in Tab. 5 shows the following retroflex acting as a blocker when a left root boundary intervenes between [ɻ] and [n]. On the other hand, the retroflex is not a blocker in *pɻa-√ɳeː-tɻ̩*, due to the coronal intervening between [n] and [ɻ]. Finally, *pɻ̩-ɳa-k-ṣi* is a case where the retroflex does not block in the absence of an intervening root boundary.

We can follow the same approach as in Sec. 3.4 to handle this complication. That is to say, we include yet another two conditional slots following the target nasal, and its mandatory adjacent sonorant. As the interval now gets exceedingly long, graphical depictions have to be broken up again across multiple lines.

(11) **IBSP interval (Version 5, Final)**



This time, open slot 3 tracks the presence of a coronal, and open slot 4 indicates whether a retroflex is present. Once again we have to forbid these segments in the neighboring fillers to ensure that if such a segment is present, it must go into one of these open slots.

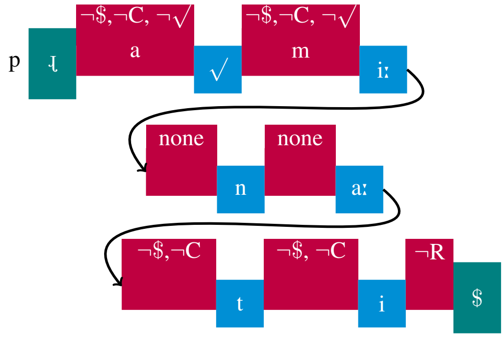
We then expand the list for forbidden 4-grams to forbidden 6-grams. The 4-gram pattern ϕnS is expanded to ϕn$S\phi'$. Just like ϕ describes the illicit segments for 1 and 2, ϕ' handles open slots 3 and 4 in (11). However, ϕ' cannot be described independently of ϕ as the relevance of slots 3 and 4 for blocking depends on the presence of a root boundary in open slot 1. Hence the options for ϕ and ϕ' have to be specified in conjunction in order to represent the conditions needed for *nati* to apply (i.e. cases where it fails to be blocked):

1	**2**	**3**	**4**
$\surd$	¬P ∧ ¬C	¬C	¬R
¬$\surd$ ∧ ¬C	¬P ∧ ¬C	¬C	¬R
¬$\surd$ ∧ ¬C	P	¬C	¬R
$\surd$	¬P ∧ ¬C	C	¬R
¬$\surd$ ∧ ¬C	¬P ∧ ¬C	C	¬R
¬$\surd$ ∧ ¬C	P	C	¬R
$\surd$	¬P ∧ ¬C	C	R
¬$\surd$ ∧ ¬C	¬P ∧ ¬C	C	R
¬$\surd$ ∧ ¬C	P	C	R

Figure 2: Open slots in $\phi \wedge \phi'$ s.t. nS is illicit

The interval in (11), with the list of forbidden 6-grams above in Figure 2, is the final version of the IBSP grammar for *nati* (although other potential variants are discussed in Sec. 4). This is a good point to reevaluate some of the earlier data points. For example, we can model some examples that illustrate conditional blocking of intervening velar/labial plosives like so:

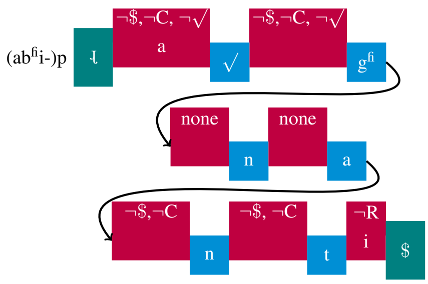
(12) **IBSP interval: pɻa√miːnaːti**



The instantiated locality domain looks quite similar to its previous iteration in (10). The main difference is that rather than having [t] and [i] in the filler following the nS sequence, those segments are pushed into the open slots that check for the presence of an anti-blocking coronal and/or blocking retroflex. The configuration of conditional slots matches $\surd$, $\neg P \wedge \neg C$, $\neg C$, $\neg R$, which is one that enforces *nati*. Consequently, the presence of an [n] in the open slot where it is forbidden causes the string to be rejected. If [n] had undergone *nati* as required, the string would not have been deemed illicit by the grammar.

The string *pɻa-*$\surd$*gʱn-an-ti*, on the other hand, is still well-formed. Even when [n] appears in the open slot, this does not yield an illicit configuration of open slots due to the presence of a root boundary in open slot 1 and a plosive in open slot 2.

(13) **IBSP interval: (abʱi-)pɻa-√gʱn-an-ti**



4 Discussion and conceptual remarks

The IBSP analysis developed over the course of Sec. 3 is with a doubt convoluted, much more so than the analysis in terms of IO-TSL. In contrast to IO-TSL, it also hinges on several idealizations that cannot be eliminated without further complicating the grammar. I will briefly sketch the most important issues here, in particular those that highlight the shortcomings of IBSP relative to IO-TSL.

At a high level of abstraction, the strategy employed in this paper boils down to a few simple tricks:

1. Furnish an open slot for every type of segment that can potentially matter for the dependency.
2. If an open slot needs to track the presence of some segment of type X, do not allow the surrounding fillers to contain X.
3. Whatever implicational relations hold between the relevant segments are compiled out into a list of forbidden k-grams.

While each step is conceptually simple, the sheer number of open slots and potential combinations of segments make proving that this approximation of *nati* is IBSP a daunting task. In addition, the first two strategies have serious drawbacks as they respectively impose a lower bound on the number of segments in the string, and an upper bound on how many segments of a specific type may occur in a specific part of the interval.

Let us consider the problem of a lower bound first. As more and more factors were incorporated into the analysis, more and more open slots had to be added to make the interval sensitive to the presence of any segments that might affect well-formedness. However, as the number of open slots grows, shorter strings are automatically considered well-formed. This is because IBSP trivially allows any string in which the interval cannot be instantiated. An interval with 6 open slots, for example, cannot be instantiated in a string that only consists of 5 symbols. In IBSP, a high number of interacting factors makes it difficult to regulate short strings.

As a remedy, Graf (2017) allows strings to be padded out by additional edge markers to enforce the required minimal length of strings. We could take a similar approach, and modify the right interval boundary to be the string edge rather than the word edge. As long as each string only represents a single phonological word rather than a string of words, the string edge is a viable replacement for the word edge. It is still far from obvious, though, that padding out can solve the problem of words where only one segment occurs between the retroflex trigger and the targeted [n]. Recall that the current interval posits two open slots, and hence at least two segments between them. While there might be some way to add even more open slots so that [n] can be "shifted" to the left and also occur in one of the first two open slots, this would render the account entirely opaque to human intuition.

In the other direction, IBSP also runs into an undesirable upper bound limit. For instance, coronals cannot go into the first or second filler, leaving only the first open slots as an option for a coronal that is somewhere to the left of [n] but not adjacent to it. If a string contains two coronals, neither one of which is adjacent to [n], the interval cannot be instantiated at all. In this case, this is unproblematic since coronals would block *nati* anyways, so either way the string is deemed well-formed. The situation is reversed, however, with coronals after [n], which undo blocking of *nati* by a retroflex. If a string contains two coronals between [n] and such a retroflex, the interval will not be instantiated and the string will incorrectly be treated as well-formed. Similarly, if more than one retroflex occurs between the sonorant following target [n] and the right interval boundary, the interval cannot evaluate the string. Again, one could fix these issues by adding more open slots and modifying the list of forbidden k-grams, but this would exacerbate the lower bound problem with short strings. It once again would make the grammar unintelligible.

Whether *nati* is actually IBSP thus cannot be answered definitively — it depends on how one generalizes from the finite data to an infinite sample. For the available data, it is certainly possible to construct the interval and the list of k-grams in a suitable manner, although it may be very difficult to verify the correctness of the analysis by hand. Once one generalizes from the data to allow an arbitrary number of coronals and retroflexes, IBSP may prove insufficient.

The latter point also holds for the intersection closure of IBSP. Suppose that each case of *nati* is given its own IBSP grammar, and that these grammars are arranged in such a fashion that the intervals for simpler cases cannot be established in the more complex cases. For instance, the interval in (5) could be amended so that the first filler may not contain a left root boundary and the last filler may not contain any retroflex. The interval then cannot be instantiated in any strings where these complicating factors are present, limiting it only to simple cases of *nati*. This solves the lower bound problem, because shorter strings are now regulated by one of the IBSP grammars for simpler cases of *nati*. At full generality, however, the upper bound problem remains. For instance, sensitivity to retroflexes requires that retroflexes may not be fillers, and thus the interval's ability to accommodate retroflexes depends on its number of open slots. As there can be only a finite number of open slots, the number of retroflexes is finitely bounded. Intersection closure can increase that bound to any desired k, but it will always be bounded. Consequently, the intersection closure of IBSP can handle the attested *nati* data, but not necessarily the most natural generalization of this data.

There are also several minor issues of data analysis, such as the status of geminates. As mentioned in Sec. 3.2, geminate [n] becomes geminate [ɳ] under *nati*. This is not captured by the current grammar, but corresponding modifications could be made. If geminate [nː] is modeled as underlying /nn/, the list of forbidden 6-grams can be modified to also block [ɳn]. Then, [ɳɳ] would be the only possible surface form. On the other hand, if [nː] is a single symbol, then the 6-grams must be modified such that [nː] is forbidden even if the following segment is not a sonorant, since the geminate acts as its own sonorant (metaphorically speaking). These are minor issues compared to the much more substantive problem of how conditional sensitivity to a segment may sometimes entail an upper bound on the number of those segments in IBSP.

For all these reasons, IBSP does not provide an insightful or elegant perspective of *nati*, in particular compared to G&M's IO-TSL treatment. Nonetheless, the IBSP of view of *nati* has identified several issues that are relevant for subregular research, most prominently the specific shortcomings of IBSP in comparison to IO-TSL. These have not been noticed before because most phonological phenomena only require sensitivity to two or three segments. We now face the question of how one should treat analyses that diverge depending on how one generalizes from the finite data sample. The intersection closure of IBSP can handle all generalizations of *nati* as long as there is an upper bound on the number of relevant segments (retroflexes, coronals, left root boundaries), whereas IO-TSL requires no such upper bounds. Which one of the two is a more appropriate characterization? It may be the case that the bounds we find in the available data are not an artifact of a finite data sample, but indicators of a principled bound to the limits of IBSP (see Joshi (2000) for a similar argument in syntax).

Finally, there is the issue of succinctness and elegance and to what extent they should be a criterion in the classification of empirical phenomena. This is a long-standing debate: if X is computationally simpler than Y, but only Y provides for a natural description, which one of the two is a better model of the relevant linguistic factors? Of course, formal language theory is well-served by having both X and Y as descriptions of the phenomenon, but if we regard subregular complexity as an abstract gauge of the cognitive machinery (cf. Rogers and Pullum, 2011), X and Y may embody vastly different claims.

5 Conclusion

I have argued that a phonotactic pattern as complex as *nati*, which can be viewed as an interaction between local and non-local dependencies with intervening material that provides blocking effects, can be modeled with the intersection closure of IBSP. However, the details depend on specific assumptions about the data, and the proposed account is fairly complicated and lacks linguistic naturalness. These drawbacks highlight specific limitations of IBSP relative to IO-TSL, and might be useful for future work on the relation between the two.

Future work could revisit my findings along two dimensions. On a formal level, it might be possible to extend IBSP grammars with mechanisms that allow for more succinct descriptions without increasing generative capacity. From a linguistic perspective, one might try to reassess the empirical status of *nati* with respect to which of its components are most natural under an IBSP-analysis. If these aspects turn out to be on empirically solid ground, this might provide indirect evidence for IBSP as a model of natural language phonotactics.

Acknowledgements

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. I would like to express my deepest gratitude to Thomas Graf, for his endless insights and patient guidance. I am thankful to my colleagues, Aniello De Santo and Alëna

Aksënova, who provided valuable feedback and are a constant source of inspiration. My appreciation is further extended to the anonymous reviewers whose comments helped improve this paper. Any errors in this work are entirely my own.

References

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Formal Grammar*, pages 35–50, Berlin, Heidelberg. Springer.

M.B. Emeneau. 1946. The nasal phonemes of Sanskrit. *Language*, page 22(2):86–93.

Thomas Graf. 2010. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics*, 16(2):Article 10.

Thomas Graf. 2017. The power of locality domains in phonology. *Phonology*, 34:385–405.

Thomas Graf. 2018. Locality domains and phonological c-command over strings. In *Proceedings of "NELS" 2017*.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Gunnar Ólafur Hansson. 2001. *Theoretical and Typological Issues in Consonant Harmony*. Ph.D. thesis, University of California, Berkeley.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological Typology, Phonetics and Phonology*, pages 126–195.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Aravind K. Joshi. 2000. Relationship between strong and weak generative power of formal systems. In *Proceedings of the Fifth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+5)*, pages 107–114, Université Paris 7.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. volume 10950 of *Lecture Notes in Computer Science*, pages 62–83. Formal Grammar, Springer, Berlin, Heidelberg.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Vischer, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. 6149:255–265.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Kevin Ryan. 2017. Attenuated spreading in Sanskrit retroflex harmony. *Linguistic Inquiry*, 48 (2):299–340.

Nazila Shafiei and Thomas Graf. 2019. The subregular complexity of syntactic islands. Ms., Stony Brook University.

William Dwight Whitney. 1889. *Sanskrit Grammar*. Oxford University Press, London.

The Rhetorical Structure of Modus Tollens: An Exploration in Logic-Mining

Andrew Potter
Computer Science & Information Systems Department
University of North Alabama
Florence, Alabama, USA
apotter1@una.edu

Abstract

A general method for mining discourse for occurrences of the rules of inference would be useful in a variety of natural language processing applications. The method described here has its roots in Rhetorical Structure Theory (RST). An RST analysis of a rule of inference can be used as an exemplar to produce a relational complex in the form of a nested relational proposition. This relational complex can be transformed into a logical expression using the logic of relational propositions. The expression can then be generalized as a logical signature for use in logic-mining discourse for instances of the rule. Generalized logical signatures reached in this manner appear to be grounded in identifiable logical relationships with their respective rules of inference. Thus, from a text, it is possible to identify a rhetorical structure, and from the structure, a relational proposition, and from the relational proposition, a generalized logical signature, and from the signature, the rule of inference residing within the text. The focus in this paper is on modus tollens and its variants, but the method is extensible to other rules as well.

1 Introduction

Recognizing occurrences of rules of inference in discourse is difficult for humans and computers alike. A method for doing so would be valuable for natural language processing, discourse analysis, and studies in logic and argumentation. Potter (2018) showed that some standard rules, including modus ponens, disjunctive syllogism, and some basic logical operations are directly accessible using Rhetorical Structure Theory (RST). This arises as a result of direct mappings between RST relations, corresponding relational propositions, and the rules of inference. For others there is no direct correspondence. This is because the rules of inference rules tend to manifest, not as individual relations, but as relational complexes, which may be embedded within deeply nested relational propositions.

This paper provides a description of a method for using RST to discover occurrences of modus tollens in natural discourse. The paper will extend this method to biconditional elimination, particularly as it relates to valid forms of denying the antecedent. Identifying relational complexes associated with these rules will support the specification of generalized logical signatures that can be used in logic-mining texts. While the method defined here is limited to modus tollens and its variants, it provides guidance for investigating other rules of inference, such as hypothetical syllogism and dilemma, and may lead to a general methodology for signature-based logic mining. This also suggests the possibility of discovering rules of inference present in discourse but not recognized in the literature of classical logic.

The approach described here presumes the availability of RST analyses, created, either interactively using tools such as O'Donnell's (1997) RSTTool or Zeldes' (2016) rstWeb, or computationally (e.g., Corston-Oliver, 1998; Hernault, Prendinger, duVerle, & Ishizuka, 2010; Pardo, Nunes, & Rino, 2004; Soricut & Marcu, 2003). These RST analyses may be restated as nested relational propositions, and these propositions can be used to generate the underlying logical organization of the text (Potter, 2018). Discovery of inference rule instantiations within this logical expression proceeds by aligning logical signatures with structural constituents of the comprehensive expression.

Lest there be any confusion as to the scope of this study, note that the objective here is not to develop a system of reasoning based on linguistic form, as in natural logic (MacCartney & Manning, 2009; Van Benthem, 1986), nor is it concerned with the logical forms of imperatives, questions, and statements, nor with the relationship between grammar and reasoning (Lakoff, 1970). The scope of this study concerns the discovery of occurrences of rules of inference as presented in discourse, as manifested in rhetorical structures, and with particular focus on modus tollens. Consistent with the fundamentals of RST, it is a logic of intended effect.

The remaining sections of this paper are as follows. First, a brief review of RST is presented using an analysis of a relevant example. This is followed by an overview of the logic of relational propositions, showing how complexes of nested relational propositions provide the basis for logical signatures useful in logic mining. Four generalized signatures for modus tollens are discussed, consisting of canonical, evidential, biconditional, and antithetical signatures. This includes a brief analysis concerning inference rule identification for incomplete relational complexes. Following this analysis is an explanation for how the logical signatures derived from discourse can be used to validate the rules of inference they serve to instantiate. The paper concludes with a discussion of the results and directions for future study. Relevant literature will be cited in passim.

2 RST Analysis of a Relevant Example

Rhetorical Structure Theory (RST) is an account of textual coherence (Mann & Thompson, 1988). It is used for describing texts in terms of the relations that hold among the text spans comprising the text. An RST relation consists of three parts: a satellite, a nucleus, and a relation. The satellite and nucleus are text spans, which are either elementary discourse units or subordinate RST relations. The distinction between satellite and nucleus arises as a result of the asymmetry of the relations. Within a relation, the nucleus is more salient than the satellite. A key consideration in defining nuclearity is the concept of locus of intended effect. The locus of intended effect may be in the nucleus, the satellite, or shared between the two. Locating the effect is important for the logical analysis of RST relations, particularly in implicative relations where the locus of intended effect will usually be the implicand (Potter, 2018).

Figure 1 shows an example of an RST analysis. The text is a short passage from J. L. Austin's translation of Frege's *Foundations of Arithmetic* (1884/1980, p. 37). The text presents an argument against the claim that numbers are merely ideas without objective reality. Frege begins by stating that he disagrees with a claim made by the mathematician Oskar Schlömilch, that numbers are ideas, not things. Frege supports his statement first by conceding that if numbers were merely ideas, then mathematics would be part of psychology. The CONDITION relation is used to indicate the dependency of the nucleus on the satellite. But this conditional is rejected using a comparison of mathematics with astronomy. This analogy is used as EVIDENCE for rejecting Schlömilch's position. That Frege's argument is an application of modus tollens

$$(((p \rightarrow q) \wedge \neg q) \rightarrow \neg p)$$

and that the RST structure presented here maps to the rule of inference may be intuitively apparent. However, as will be developed in this paper, this need not, and in most cases cannot, be merely a matter of intuition.

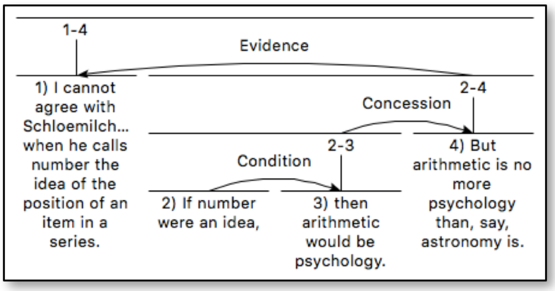


Figure 1: RST Analysis of Frege's Argument Against Psychologism

3 The Logic of Relational Propositions

It has been argued that Rhetorical Structure Theory is incapable of representing inferential patterns, because argumentative and rhetorical relations are said to be orthogonal to one another, and because RST relations provide little or no indication of alignment with the rules of inference (Budzynska, Janier, Reed, & Saint-Dizier, 2016). However, the structure of an RST analysis reflects the structure of its argument. EVIDENCE is evidential, MOTIVATION is motivational, and ENABLEMENT is enabling. This would suggest the logic and reasoning are not too far below the surface. As shown by Potter (2018), for any RST relation there is a corresponding logical form, and these forms combine to construct logical expressions that map to RST tree structures and serve as logical interpretations of the organization of a text. The approach used for deriving these interpretations is based on discourse entities known as relational propositions. Relational propositions are implicit assertions that arise between clauses within a text and are essential to the effective functioning of the text (Mann & Thompson, 1986a, 1986b, 2000). RST and relational propositions provide parallel accounts of discourse coherence. While RST identifies structures of coherence relations among the spans within a text, relational propositions treat these relations as implicit relational acts that account for how the text functions (Mann & Thompson, 1986b).

A relational proposition consists of a predicate and a pair of discourse units. The predicate corresponds to the RST relation, and the units correspond to the satellite and nucleus. In this paper relational propositions are specified using a functional notation. This permits concise representation of nested relational propositions. For example, the relational proposition for the RST analysis of the Frege argument shown in Figure 1 is as follows:

evidence(concession(condition(2,3),4),1)

where each elementary discourse unit is identified numerically in order of appearance in the text. Each relational predicate is associated with a logical form. In the above relational proposition, the *condition* predicate is defined as material implication, $(s \rightarrow n)$. The satellite materially implies the nucleus. Granted, there are persuasive arguments in favor of treating *condition* as biconditional (e.g., Geis & Zwicky, 1971; Horn, 2000; Karttunen, 1971; Moeschler, 2018; van der Auwera, 1997a, 1997b); however, for the purpose of logic mining the biconditional interpretation of *condition* will frequently be unnecessary, and preserving the distinction conditional and biconditional can be a useful.

With the CONCESSION predicate, the writer acknowledges a perceived incompatibility between the situations presented in the satellite and nucleus and uses this acknowledgement to forestall objections that might otherwise have arisen as a result of the perceived incompatibility. By pre-empting the objection, the writer smooths the way to increasing the reader's positive regard for the situation presented in the nucleus. Logically then, we can say that it is not the case that the satellite provides grounds for rejecting the nucleus: $\neg(s \rightarrow \neg n)$. Upon neutralizing this objection, the writer further invites the reader to infer from this the claim presented by the nucleus. The reasoning thus becomes an instance of modus ponens in which the condition of the major premise is a negated conditional statement:

$$(((\neg(s \rightarrow \neg n) \rightarrow n) \land \neg(s \rightarrow \neg n)) \rightarrow n)$$

With the EVIDENCE predicate, the satellite provides evidence in support of the nucleus. For the relation to achieve its intended effect, the reader must accept the satellite and recognize its implicative relationship with the nucleus. If the antecedent is believable, the consequent will also be believable. To achieve its effect, EVIDENCE requires that the antecedent (i.e. the satellite) be asserted. Hence the logical form of EVIDENCE is modus ponens:

$$(((s \rightarrow n) \land s) \rightarrow n)$$

The three logical forms (*condition*, *concession*, and *evidence*), corresponding to the relations used in the Frege analysis, can be used to construct the logical expression of the nested relational proposition, which expands to the following valid argument:

$$((((((\neg((2 \rightarrow 3) \rightarrow \neg 4) \rightarrow 4) \land \neg((2 \rightarrow 3) \rightarrow \neg 4)) \rightarrow 4) \rightarrow 1) \land (((\neg((2 \rightarrow 3) \rightarrow \neg 4) \rightarrow 4) \land \neg((2 \rightarrow 3) \rightarrow \neg 4)) \rightarrow 4)) \rightarrow 1)$$

Using this technique, it is possible to generate logical expressions for any RST analysis. While the resulting expressions can be complex, they are constructed from the simple logical forms defined for each of the relational predicates. As will be detailed in Section 4, these forms are generalizable as logical signatures that may be used in mining texts for occurrences of rules of inference.

Note that discourse units used in relational propositions need not be truth-functional in the restrictive sense of the term. Although it is common practice present logic in terms of truth values and truth functions, these semantics are arbitrary, and we could just as well speak of on and off, + and -, 1 and 0, yes and no, open and closed, satisfiability and unsatisfiability, or any other bivalent conceptualization, including belief and disbelief, positive and negative regard, desire and indifference, interest and disinterest, understanding and misunderstanding, or ability and inability. To the extent that the primitives of RST can be understood in terms of bivalent values, they are amenable to logical treatment.

4 Relational Complexes

As noted earlier, some inference rules manifest as single relational predicate, but this is not always the case. Modus tollens requires multiple predicates, and these predicates may be combined in various ways. Each of these combinations, for any given instance, is a relational complex. A relational complex may then be generalized and normalized to create a signature, or logical pattern that may then be used to locate other instances of the rule in discourse.

The generalization process consists in replacing the numeric unit identifiers with normalized alphabetic variables. Normalization consists in identifying discourse units that are sufficiently similar semantically to indicate material equivalence or negation. This paper makes no attempt to define a technology for measuring semantic textual similarity. There are already numerous research efforts in that area. For example, Finch, Hwang, and Sumita (2005) repurposed machine translation evaluation methods to determine sentence-level semantic equivalence, Tsatsaronis, Varlamis, and Vazirgiannis (2010) developed a measure of semantic relatedness which capitalizes on a word-to-word semantic relatedness measure and extended it to measure the relatedness between texts, and Sultan, Bethard, and Sumner (2015) developed supervised and unsupervised systems for measuring sentence similarity. Addressing negation detection, Basile, Bos, Evang, and Venhuizen (2012) used discourse representation structures for negation detection, and Harabagiu, Hickl, and Lacatusu (2006) interpreted negation using a combination of overt and indirectly licensed negation. For the present study, normalizations are hand-crafted. Thus, for the generalized signature

$$((((((\neg((p \to q) \to \neg r) \to r) \wedge \neg((p \to q) \to \neg r)) \to r) \to s) \wedge (((\neg((p \to q) \to \neg r) \to r) \wedge \neg((p \to q) \to \neg r)) \to r)) \to s)$$

the normalized logical form, with double negations removed is:

$$((((((\neg((p \to q) \to q) \to \neg q) \wedge \neg((p \to q) \to q)) \to \neg q) \to \neg p) \wedge (((\neg((p \to q) \to q) \to \neg q) \wedge \neg((p \to q) \to q)) \to \neg q)) \to \neg p)$$

We can use this logical form as a template for identifying comparable relational propositions within texts, keeping in mind that any of the elements of the expression may refer recursively to lower level complex expressions. To the extent that the comparisons align, the logical expressions for each relational proposition will comprise the sought-after relational complexes, which provide the basis for the logical signature.

5 Canonical Modus Tollens

Modus tollens is a valid argument of the form:

$$(((p \to q) \wedge \neg q) \to \neg p)$$

The categorical premise $(\neg q)$ denies the consequent of the conditional premise, implying the negation of the antecedent $(\neg p)$. Figure 2 shows an RST analysis of a *Wikipedia* example of modus tollens. As shown, the writer concedes that the conditional relationship between Rex as a

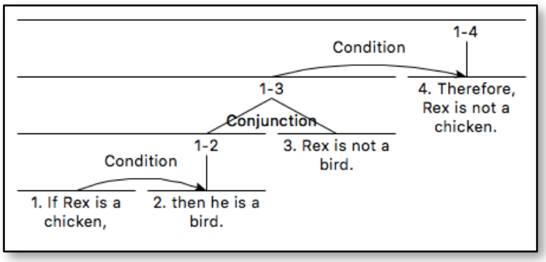


Figure 2: Rhetorical Structure of Modus Tollens

chicken and Rex as a bird holds, but rejects the proposition that he is a bird. From this, we may reason, Rex is no chicken. The relational proposition for this structure is

condition(conjunction(condition(1,2),3),4)

And the relational complex for this proposition therefore is:

$$(((1 \to 2) \wedge 3) \to 4)$$

This may be generalized and normalized to

$$(((p \to q) \wedge \neg q) \to \neg p)$$

which is modus tollens. Stated canonically, the RST relations are subject matter, rather than presentational, because there is no intent to influence an inclination in the reader. In practice, however, modus tollens is commonly used as an act of persuasion. This leads to the evidential and antithetical signatures for modus tollens.

6 Evidential Modus Tollens

When the writer uses modus tollens with the intent to influence the reader's beliefs, the EVIDENCE relation may be employed. This intended effect adds to the complexity of the logical structure of the argument. This occurs in Frege's argument against the claim that numbers are merely ideas without objective reality, introduced earlier. Frege's argument, shown in Figure 1, relies on modus tollens for its validity. EVIDENCE is used to link the argument's premises to the conclusion. As specified by the definition of modus tollens, the argument starts with a conditional premise:

If number were an idea, then arithmetic would be psychology,

followed by a categorical premise that denies the consequent of the conditional premise,

But arithmetic is no more psychology than, say, astronomy is,

and a conclusion that infers the denial of the antecedent of the conditional premise:

I cannot agree with Schloemilch…when he calls number the idea of the position of an item in a series.

The relational proposition for the Frege analysis,

evidence(concession(condition(p,q),r),s)

generalizes to the logical expression:

$$((((((\neg((p \rightarrow q) \rightarrow q) \rightarrow \neg q) \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q) \rightarrow \neg p) \wedge (((\neg((p \rightarrow q) \rightarrow q) \rightarrow \neg q) \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q)) \rightarrow \neg p)$$

Any analysis that matches this generalized signature will be an instantiation of the modus tollens rule of inference. That this is so is supported in part by the signature's derivation from an exemplar of modus tollens, and is further supported, as will be discussed in detail in Section 9, by the realization that the rule of inference is deducible from the signature. That is to say, for any such argument, the canonical rule is logically implicit within the RST analysis, and therefore within the text.

7 Biconditional Modus Tollens

The CONDITION relation sometimes represents a biconditional logical relation. This is apparent in part from the definition of the relation as specified by Mann and Thompson (1987), that realization of the situation presented in the nucleus (the consequent) *depends* upon the realization of the situation presented in the satellite (the antecedent), and it is also observable in the text they used as their example of the relation:

N: Employees are urged to complete new beneficiary designation forms for retirement or life insurance benefits

S: whenever there is a change in marital or family status.

A *change in marital or family status* is the condition under which employees are urged to complete new beneficiary designation forms. The reader recognizes that the realization of the nucleus

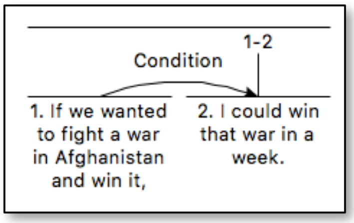


Figure 3: Counterfactual Modus Tollens

depends on the realization of satellite. If there is no change in status, there is no need to complete new forms. If the satellite remains unrealized, so will the nucleus. Thus, the relation is biconditional ($s \leftrightarrow n$).

Occurrences of the biconditional as modus tollens may employ the counterfactual in the antecedent. The counterfactual contains the denial of the antecedent within the antecedent itself. In the example shown in Figure 3, Donald Trump argues that if he wanted to win the war in Afghanistan, he could do so within a week. The counterfactuality of the antecedent indicates that he does not wish to do so, with the implication that we therefore cannot do so. This interpretation leads to a relational proposition defined not only on the basis of the explicit rhetorical structure, but the implicit relations as well:

condition(conjunction(condition(1,2),[3]),[4])

which normalizes to the biconditional modus tollens: $(((p \leftrightarrow q) \wedge \neg p) \rightarrow \neg q)$. When the normalization process indicates denial of the antecedent, the charitable interpretation will be that the CONDITION relation is being used as biconditionally. Not only may the denial of the antecedent be implicit, the consequent itself may be implicit. Incomplete conditionals such as

1. *If only Miss Hawkins would get a job…*

have an implicit implicative potentiality. While this example leaves much to the reader's imagination, with assistance from context provided by the writer, or from the reader's world knowledge (Elder & Savva, 2018), a pragmatic conjecture such as

2. *[then surely her situation would be improved.]*
3. *[But, alas, she has not gotten a job.]*
4. *[And so her situation remains unimproved.]*

seems plausible, and results in the relational complex:

cause(concession(condition(1,[2]),[3]),[4])

As constructed, the inference relies on denying the antecedent. Hence it is another example of biconditional modus tollens. However, the logic differs from the previous example, due to the use of the *cause* predicate instead of *condition*. The *cause* predicate has the same logical form as *evidence*, and as such is used to link the argument's premises to the conclusion. Clearly, however, the more fragmentary the information, the greater the risks of conjecture, and the greater risk of false positives.

8 Antithetical Modus Tollens

ANTITHESIS is used as part of a modus tollens relational complex in a manner rhetorically similar to CONCESSION. This is perhaps owing to the similarity of the two relations (Stede, 2008). In the example shown in Figure 4, the structure follows the familiar pattern of modus tollens, but now the

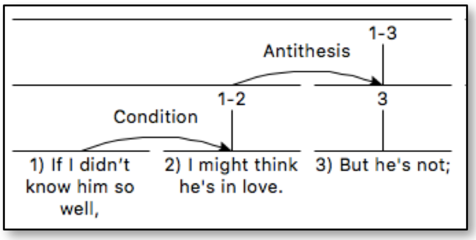


Figure 4: ANTITHESIS as Modus Tollens

CONDITION is a satellite of the ANTITHESIS rather than of CONCESSION. The logical form, and hence the signature, is disjunctive syllogism,

$$((((p \rightarrow q) \vee \neg q) \wedge \neg(p \rightarrow q)) \rightarrow \neg q)$$

Thus ANTITHESIS, when the satellite is conditional, is modus tollens. Alternatively, the CAUSE relation may be used as satellite to the ANTITHESIS relation, as shown in Figure 5. This text is interesting in several respects. From the logical perspective, there are arguments within arguments such that the consequences of one become the condition of another. And counterfactual conditionality is used to implement a strategy of reductio ad absurdum, such that the conclusion of the text indicates its own negation. Logic mining is useful in sorting this out. The text divides conveniently into two parts. Units 1-3 implement the causal variety of antithetical modus tollens:

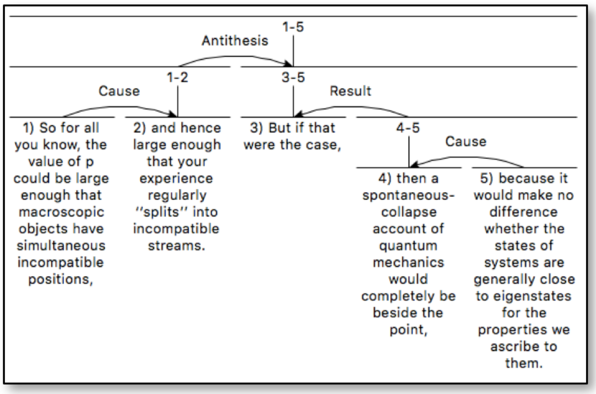


Figure 5: The Cause-Antithesis Modus Tollens

$$((((((p \rightarrow q) \wedge p) \rightarrow q) \vee r) \wedge \neg(((p \rightarrow q) \wedge p) \rightarrow q)) \rightarrow r)$$

That this is an occurrence of antithetical modus tollens can be realized by evaluating the causal argument to obtain its result, q, so that the expression becomes

$$(((q \vee r) \wedge \neg q) \rightarrow r)$$

which when normalized becomes a signature for antithetical modus tollens:

$$(((p \vee q) \wedge \neg p) \rightarrow q)$$

As discussed below in Section 9, modus tollens is provable using disjunctive syllogism. An alternative approach would be to realize that if $((p \rightarrow q) \wedge p)$, as indicated by CAUSE, then the CONDITION $(p \rightarrow q)$ holds as well. The same approach can be used for segments 3-5. The if-then statement of 3-4 is coded as a RESULT, because it is the antecedent of the condition that is salient in this text. Segment 3, or *r*, situated conditionally within the argument, is the negation of "that is not the case." The consequent, provided in 4-5, provides the reductio ad absurdum. That is, if "that were the case," untenable results would follow.

9 The Significance of Signatures

The question will arise as to the significance of logical signatures. Are they grounded in identifiable logical relationships with their respective rules of inference, or is the correspondence between signatures and rules simply a happy coincidence? Both signatures and rules are valid arguments, both share the same elementary propositions, and both reach the same conclusion. It would therefore be useful to determine whether the rules of inference are deducible from the signatures, and if not, what the nature of the relationship is. So now we can examine each the signatures introduced above and determine their relationship to modus tollens. The signatures to be considered include canonical, evidential, biconditional, and antithetical modus tollens. For *canonical modus tollens*, the signature maps directly to the inference rule; it is indeed simply a statement of the rule, $(((p \rightarrow q) \wedge \neg q) \rightarrow \neg p)$. *Evidential modus tollens* is a more interesting case. It has already been shown that the logical signature for

evidence(concession(condition(p,q),r),s)

is

$$((((((\neg((p \rightarrow q) \rightarrow q) \rightarrow \neg q) \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q) \rightarrow \neg p) \wedge (((\neg((p \rightarrow q) \rightarrow q) \rightarrow \neg q) \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q)) \rightarrow \neg p)$$

This expression contains two occurrences of the valid argument

$$(\neg((p \rightarrow q) \rightarrow q) \rightarrow \neg q)$$

We evaluate and replace those occurrences with their consequent, $\neg q$, resulting in

$$(((((\neg q \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q) \rightarrow \neg p) \wedge ((\neg q \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q)) \rightarrow \neg p)$$

which contains two occurrences of the valid argument

$$((\neg q \wedge \neg((p \rightarrow q) \rightarrow q)) \rightarrow \neg q)$$

for which we also substitute the consequent, $\neg q$, resulting in the valid argument

$$(((\neg q \rightarrow \neg p) \wedge \neg q) \rightarrow \neg p)$$

for which the implicant

$$((\neg q \rightarrow \neg p) \wedge \neg q)$$

is materially equivalent to the implicant of modus tollens:

$$((\neg q \rightarrow \neg p) \wedge \neg q) \leftrightarrow ((p \rightarrow q) \wedge \neg q)$$

Thus the evidential interpretation effectively reduces to modus tollens. This is applicable to the logical forms of each of the modus ponens presentational relations, including BACKGROUND, ENABLEMENT, EVIDENCE, JUSTIFY, MOTIVATION, and PREPARATION, as well as the causal relations.

The presentational version of *biconditional modus tollens* operates similarly. The relational proposition

evidence(concession(condition(p,q),r),s)

normalizes to

$$((((((\neg((p \leftrightarrow q) \rightarrow p) \rightarrow \neg p) \wedge \neg((p \leftrightarrow q) \rightarrow p)) \rightarrow \neg p) \rightarrow \neg q) \wedge (((\neg((p \leftrightarrow q) \rightarrow p) \rightarrow \neg p) \wedge \neg((p \leftrightarrow q) \rightarrow p)) \rightarrow \neg p)) \rightarrow \neg q)$$

The modus ponens

$$(((\neg((p \leftrightarrow q) \rightarrow p) \rightarrow \neg p) \wedge \neg((p \leftrightarrow q) \rightarrow p)) \rightarrow \neg p)$$

occurs twice within this expression. Replacing this with its consequent, $\neg p$, yields

$$(((\neg p \rightarrow \neg q) \wedge \neg p) \rightarrow \neg q)$$

which is modus tollens. This is applicable to biconditional occurrences of the same RST relations as evidential modus tollens, except that the categorical premise normalizes to the negation of the antecedent of the conditional premise, rather than the consequent. It is by this means that this biconditional modus tollens can be distinguished from evidential modus tollens.

The relational proposition of *antithetical modus tollens* is *antithesis(condition(p,q),r)* for which the generalized signature is

$$((((p \rightarrow q) \vee \neg q) \wedge \neg(p \rightarrow q)) \rightarrow \neg q)$$

Since one of the proofs of modus tollens is based on disjunctive syllogism, it can be shown that modus tollens follows from the normalized expression. The major premise of the disjunctive syllogism, $((\neg p \vee q) \wedge \neg q)$, implies $(p \rightarrow q)$, so that if it is the case that

$$(((\neg p \vee q) \wedge \neg q) \rightarrow (p \rightarrow q))$$

it follows that both the premise and the conclusion hold,

$$(((\neg p \vee q) \wedge \neg q) \wedge (p \rightarrow q))$$

and it is a tautology that

$$(((\neg p \vee q) \wedge \neg q) \wedge (p \rightarrow q)) \leftrightarrow ((p \rightarrow q) \wedge \neg q)$$

Thus, modus tollens may be inferred from the logical signature for *antithesis(condition(p,q),r)*. And thus, the evidential, biconditional, and antithetical signatures can be used, not only to discover instances of modus tollens in discourse, they are grounded in the rule of inference they are designed to detect.

10 Conclusion

This exploration of modus tollens has shown how relational propositions can be used to support discourse logic-mining using logical signatures as a means for discovering occurrences of standard rules of inference in discourse. In addition to modus tollens, several other signatures that serve as indicators of rules of inference have been noted. EVIDENCE and other pragmatic and causal relations map directly to modus ponens, and ANTITHESIS implements disjunctive syllogism. Further research is needed to determine what additional signatures can be identified. These would provide a rich set of resources for logic-mining discourse and reduce the need for ad hoc procedures for inference rule identification and would eventually support a greater capability for automated analysis.

Automated identification of inference rules within discourse would require development and integration of several capabilities. Although there has been significant work in automated detection of RST relations (e.g., Corston-Oliver, 1998; Hernault et al., 2010; Pardo et al., 2004; Soricut & Marcu, 2003), such a capability would need to generate output as nested relational propositions of complex structures. Prototype software already exists for generating logical expressions from nested relational propositions of arbitrary size and complexity (Potter, 2018). A unification algorithm could be used for identifying instantiations of inference rules in nested relational propositions. The generalized signatures would subsume instances of inference rules in a relational proposition. Subsumption would succeed when the proposition contains a logical structure isomorphic with the signature. The signature would need to match both simple and composite spans, so that instantiation could occur at any level within the structure.

Using RST as the starting point for inference rule discovery simplifies the task, but also delimits it. These delimitations arise not so much the result of well-known concerns about the validity of RST (e.g., Asher & Lascarides, 2003; Budzynska et al., 2016; Grosz & Sidner, 1986; Knott, Oberlander, O'Donnell, & Mellish, 2001; Moore & Pollack, 1992; Sanders, Spooren, & Noordman, 1992; Webber, Stone, Joshi, & Knott, 2003; Wiebe, 1993; Wolf & Gibson, 2005), but out of a fundamental feature of the theory—namely that it is a theory of coherence relations. Perhaps this delimitation is an asset. By basing the concept of logic-mining on a theory of coherence relations, it is by definition constrained to discursive inferences discoverable within a text. The granularity of analysis being at the clausal level, the inferences discoverable among these clauses are propositional. A benefit of this is that many problems in natural language inferencing, such as those described by Lakoff (1970), van Benthem (2008), MacCartney (2009) and Karttunen (2015), e.g., determining logical relationships among arbitrarily selected assertions, are avoided. They are avoided not because they do not exist, for indeed they do, but because they need not come to the surface. A practical solution for logic-mining texts for rules of inference should be both useful and interesting, and perhaps the techniques arising from this work will contribute to solving grander challenges. For now, the essence of logic-mining is that from a text, it is possible to identify a rhetorical structure, and from the structure, a relational proposition, and from the relational proposition, a generalized logical signature, and from the signature, the rule of inference residing within the text.

References

Asher, N., & Lascarides, A. (2003). *Logics of conversation*. Cambridge, UK: Cambridge University Press.

Budzynska, K., Janier, M., Reed, C., & Saint-Dizier, P. (2016). Theoretical foundations for

illocutionary structure parsing. *Argument & Computation, 7*(1), 91-108.
Corston-Oliver, S. H. (1998). *Computing representations of the structure of written discourse.* University of California, Santa Barbara, CA.
Elder, C.-H., & Savva, E. (2018). Incomplete conditionals and the syntax-pragmatics interface. *Journal of Pragmatics, 138*, 45-59.
Frege, G. (1884/1980). *The foundations of arithmetic* (J. L. Austin, Trans. 2nd Revised ed.). Evanstown, Ilinois: Northwestern University Press.
Geis, M. L., & Zwicky, A. M. (1971). On invited inferences. *Linguistic lnquiry, 2*, 561-566.
Grosz, B., & Sidner, C. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics, 12*(3), 175-204.
Hernault, H., Prendinger, H., duVerle, D. A., & Ishizuka, M. (2010). HILDA: A Discourse parser using Support Vector Machine classification. *Dialogue and Discourse, 1*(3), 1-33.
Horn, L. R. (2000). From if to iff: Conditional perfection as pragmatic strengthening. *Journal of Pragmatics, 32*(3), 289-326.
Karttunen, L. (1971). Counterfactual conditionals. *Linguistic Inquiry, 2*(4), 566-569.
Karttunen, L. (2015). From natural logic to natural reasoning. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing* (pp. 295-309). Cham: Springer International Publishing.
Knott, A., Oberlander, J., O'Donnell, M., & Mellish, C. (2001). Beyond elaboration: The interaction of relations and focus in coherent text. In T. Sanders, J. Schilperoord, & W. Spooren (Eds.), *Text Representation: Linguistic and Psycholinguistic Aspects* (pp. 181-196). Amsterdam: John Benjamins.
Lakoff, G. (1970). Linguistics and natural logic. *Synthese, 1/2*, 151-271.
MacCartney, B. (2009). *Natural language inference.* Stanford University, Stanford.
MacCartney, B., & Manning, C. D. (2009). An extended model of natural logic. *Proceedings of the Eighth International Conference on Computational Semantics* (pp. 140-156). Tilburg, The Netherlands: Association for Computational Linguistics.
Mann, W. C., & Thompson, S. A. (1986a). Assertions from discourse structure. *HLT '86: Proceedings of the workshop on strategic computing natural language* (pp. 257-270). Morristown, NJ: Association for Computational Linguistics.
Mann, W. C., & Thompson, S. A. (1986b). Relational propositions in discourse. *Discourse Processes, 9*(1), 57-90.
Mann, W. C., & Thompson, S. A. (1987). *Rhetorical structure theory: A theory of text organization* (ISI/RS-87-190). Marina del Rey, CA: University of Southern California, Information Sciences Institute (ISI).
Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text, 8*(3), 243-281.
Mann, W. C., & Thompson, S. A. (2000). *Toward a theory of reading between the lines: An exploration in discourse structure and implicit communication.* Paper presented at the Seventh International Pragmatics Conference, Budapest, Hungary.
Moeschler, J. (2018). On the pragmatics of logical connectives: Are connectives truth-functional? In D. V. Olmen, T. Mortelmans, & F. Brisard (Eds.), *Aspects of linguistic variation* (pp. 207-232). Berlin: De Gruyter.
Moore, J. D., & Pollack, M. E. (1992). A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics, 18*(4), 527-544.
O'Donnell, M. (1997). RST-Tool: An RST analysis tool. *Proceedings of the 6th European Workshop on Natural Language Generation*. Duisburg, Germany: Gerhard-Mercator University.
Pardo, T. A. S., Nunes, M. d. G. V., & Rino, L. H. M. (2004). DiZer: An automatic discourse analyzer for Brazilian Portuguese. *Advances in Artificial Intelligence – SBIA 2004 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings*. Berlin: Springer.
Potter, A. (2018). Reasoning between the lines: A logic of relational propositions. *Dialogue and Discourse, 9*(2), 80-110.
Sanders, T. J. M., Spooren, W. P. M., & Noordman, L. G. M. (1992). Toward a taxonomy of coherence relations. *Discourse Processes, 15*, 1-35.
Soricut, R., & Marcu, D. (2003). *Sentence level discourse parsing using syntactic and lexical information.* Paper presented at the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), Edmonton, Canada.

Stede, M. (2008). Disambiguating rhetorical structure. *Research on Language and Computation, 6*(3), 311-332.
Van Benthem, J. (1986). Natural logic. In J. Van Benthem (Ed.), *Essays in logical semantics* (pp. 109-119). Dordrecht: Springer Netherlands.
van Benthem, J. (2008). A brief history of natiural logic. *Logic, Navya-Nyāya & applications: Homage to Bimal Krishna Matilal* (pp. 21-42). Lomdon: College Publications.
van der Auwera, J. (1997a). Conditional perfection. In A. Athanasiadou & R. Dirven (Eds.), *On conditionals again* (pp. 169-190). Amsterdam: Benjamins.
van der Auwera, J. (1997b). Pragmatics in the last quarter century: The case of conditional perfection. *Journal of Pragmatics, 27*(3), 261-274.
Webber, B. L., Stone, M., Joshi, A., & Knott, A. (2003). Anaphora and discourse structure. *Computational Linguistics, 29*(4), 545-587.
Wiebe, J. M. (1993). Issues in linguistic segmentation. *Workshop on Intentionality and Structure in Discourse Relations, Association for Computational Linguistics* (pp. 148-151). Columbus, Ohio: Association for Computational Linguistics.
Wolf, F., & Gibson, E. (2005). Representing discourse coherence: A corpus-based analysis. *Computational Linguistics, 31*(2), 249-287.
Zeldes, A. (2016). rstWeb – A browser-based annotation interface for Rhetorical Structure Theory and discourse relations. *Proceedings of NAACL-HLT 2016 (Demonstrations)* (pp. 1-5). San Diego, California: Association for Computational Linguistics.

Appendix. Texts Cited

Forgey, Q. (2019, July 22). Trump: I could win Afghanistan war 'in a week'. *Politico.* Retrieved from https://www.politico.com/story/2019/07/22/trump-afghanistan-war-1425692
Frege, G. (1884/1980). *The foundations of arithmetic* (J. L. Austin, Trans. 2nd Revised ed.). Evanstown, Ilinois: Northwestern University Press.
Goodman, L. (2014). *Indefensible: A novel*. New York: Atria.
Lewis, P. J. (2003). Quantum mechanics and ordinary language: The fuzzy link. *Philosophy of Science, 70*(5), 1437-1446.
Stead, C. (1945). *For love alone*. New York: Harcourt.
Wikipedia contributors. (2019, July 23). Modus tollens. *Wikipedia, The Free Encyclopedia.* Retrieved from https://en.wikipedia.org/w/index.php?title=Modus_tollens

Unsupervised Formal Grammar Induction with Confidence

Jacob Collard
Cornell University
jacob@thorsonlinguistics.com

Abstract

I present a novel algorithm for minimally supervised formal grammar induction using a linguistically-motivated grammar formalism. This algorithm, called the Missing Link algorithm (ML), is built off of classic chart parsing methods, but makes use of a probabilistic confidence measure to keep track of potentially ambiguous lexical items. Because ML uses a structured grammar formalism, each step of the algorithm can be easily understood by linguists, making it ideal for studying the learnability of different linguistic phenomena. The algorithm requires minimal annotation in its training data, but is capable of learning nuanced data from relatively small training sets and can be applied to a variety of grammar formalisms. Though evaluating an unsupervised syntactic model is difficult, I present an evaluation using the Corpus of Linguistic Acceptability and show state-of-the-art performance.[1]

1 Introduction

Most research on learning algorithms for natural language syntax has focused on supervised parsing, in which the parser learns from sentences in the target language paired with a corresponding, hand-constructed parse tree. Major natural language corpora, such as the Penn Treebank (Marcus et al., 1994) and the Universal Dependencies framework (Nivre et al., 2016) exemplify this tendency. This has allowed for highly performant models for dependency parsing such as ClearNLP (Choi and McCallum, 2013), CoreNLP (Manning et al., 2014), Mate (Bohnet, 2010), and Turbo (Martins et al., 2013), all of which have achieved an accuracy of over 89% on standard evaluation tasks (Choi et al., 2015).

[1]In the interest of reproducibility, the code used to generate these results is provided at https://github.com/thorsonlinguistics/scil2020

Unsupervised learning for natural language processing is a much more difficult task, as the algorithm must explore the entire search space with minimal confirmation of its hypotheses. Nevertheless, a number of algorithms have attempted to solve the problem of unsupervised parsing. Most of these rely on gold standard part of speech tags (Headden III et al., 2009; Spitkovsky et al., 2010), though there are some exceptions, such as Spitkovsky (2011). Almost all unsupervised algorithms for natural language syntactic processing are based on dependency parsing; most of the published literature on other grammar formalisms, such as tree-adjoining grammar (TAG) and combinatory categorial grammars (CCG) is either supervised or hand-engineered. Again, there are some exceptions, such as (Bisk et al., 2015), which learns CCGs using a small amount of initial part-of-speech data. Edelman et al. (2003) also present a model of unsupervised learning which blends properties of construction grammars with tree-adjoining grammars; however, their model has not, as yet, been evaluated empirically.

Other models are only indirectly supervised; the syntax of the target language is learned without any syntactic annotations, but annotations may be present representing other facts about the sentence, such as its logical form. Notable examples of this include work by Kwiatkowski et al. (2010; 2011) and Artzi and Zettlemoyer (2013).

Another recent innovation in unsupervised learning is the introduction of pre-trained language models for deep learning algorithms, such as BERT (Devlin et al., 2018) and its relatives. These algorithms can be pre-trained on raw text in order to produce a language model which can then be used to bootstrap learning for a wide variety of additional tasks. Though supervision may be required by these downstream tasks, the unsupervised component has been shown to greatly improve learning. The representations that these models produce are somewhat opaque; though they have been shown to represent syntactic information for some tasks (Goldberg, 2019), an exact description of what the model is representing is difficult to produce.

Though most of the above systems do not rely on a strict notion of grammar formalism, in this paper, I will argue that a well-defined grammar formalism can produce strong results when used as the basis for an unsupervised learning algorithm. Dependence on a grammar formalism has a number of benefits. First, it means that each step of the algorithm can be (relatively) easily understood by humans. Each processing step either produces a novel derivation for a sentence or reinforces an old one, and each derivation conforms to the rules of the given formalism. Thus, as long as the rules of the formalism are understood, the meaning behind each processing step can also be understood. Second, using a grammar formalism ensures that certain facts about the resulting grammar will always hold. For example, using CCG or TAG will guarantee that the resulting grammar is in the class of mildly context-sensitive languages. Third, using a grammar formalism means that the properties of the grammar formalism can be studied as well. Though formalisms such as CCG and TAG are weakly equivalent (Joshi et al., 1990), there may be differences between the two formalisms with respect to learning. Similarly, different variants of a particular formalism can be studied as well. For example, different combinators can be added or removed from CCG to produce different learning results. By using the grammar formalism as a core parameter in learning, the formalism becomes an independent variable that can be explored.

In this paper I introduce an algorithm, called the Missing Link algorithm (ML), which has several interesting properties which, I argue, are beneficial to the study of linguistics, grammar formalisms, and natural language processing. These properties include:

- **Minimal supervision.** The Missing Link algorithm learns from raw, tokenized text. The only annotation required is assurance of the sentencial category, which is trivial for most training sets and grammar formalisms.
- **Formalism Dependence.** The grammar formalism is the core motivator for learning and parsing in Missing Link. This means that the formalism can easily be replaced with another and that the formalism can be studied as a parameter.
- **Interpretability.** Due partly to formalism dependence, the Missing Link algorithm is highly interpretable. Each step of the algorithm can be viewed as a derivation using the input formalism.
- **Performance.** The Missing Link algorithm performs well on an evaluation using linguistic acceptability judgments. The results of the evaluation are competitive with supervised algorithms such as BERT for the specific task used. Missing Link supplements the input formalism with a model of confidence for lexical entries that allows it to robustly handle potential ambiguity.

1.1 Related Work

The Missing Link algorithm builds off of relatively simple models for grammar induction and parsing. Parsing is done via a simple bottom-up chart-based method (Younger, 1967; Kasami, 1965). Learning is done in a top-down fashion using the same chart, with some extensions described in Section 2.2.

The Missing Link algorithm is closely related to the Unification-Based Learning (UBL) algorithm described in Kwiatkowski et al. (2010). UBL is an algorithm for semantic parsing, but the decomposition operations used in Missing Link are essentially the same as the higher-order unification used in UBL, albeit applied directly to syntactic categories instead of logical forms. Unlike UBL, Missing Link ensures that every stop of processing is interpretable; the probabilistic grammar used in UBL can potentially obscure why individual parses are excluded.

2 The Missing Link Algorithm

There are two main stages to the Missing Link algorithm: parsing and learning, which are performed in order for every sentence in the training set. As the algorithm processes more sentences, it updates a lexicon, mapping words to their syntactic categories and a probability representing how confident the algorithm is that the given category is valid in the target grammar.

2.1 Inputs

The core inputs to the Missing Link algorithm are:

- A grammar formalism, which defines a (possibly infinite) set of grammatical units E and two functions: COMPOSE : $E \times E \to E*$ and DECOMPOSE : $E \times E \times E \to (E \times E)*$. E always contains a special null element 0 indicating that the grammatical category is not known.

- A collection of training examples. Each training example consists of a tokenized sentence and an annotation describing the possible grammatical categories of the sentence.

The two functions of the grammar formalism determine the behavior of the parser and learner. The COMPOSE function returns a collection of elements in E that can be produced by combining the two input elements. This typically represents the basic structure-building operation of the given formalism. In Minimalism, for example, it corresponds to MERGE; in CCG, it corresponds to the various combinators; in TAG to substitution and adjunction, etc.

The DECOMPOSE function is essentially the inverse of COMPOSE. It returns the set of pairs of elements that can be composed to produce a given input. Though it takes three elements, only the first, representing the root, is necessary. The second arguments are supplied to force one or both of the elements in the results to take a particular value. This will become important to avoid exploring the entirety of the search space; when a value is already known, the DECOMPOSE function will maintain that value whenever possible.

Note that the input of Missing Link places some restrictions on the types of formalisms that can be used. In particular, the formalism does not carry any language-specific information outside of the lexicon – the formalism must be strongly lexicalized. Many major formalisms, including CCG and TAG, adhere to this rule, though some formalisms, such as standard context-free grammars, cannot be represented by Missing Link. In addition, the results of DECOMPOSE and COMPOSE must be finite sets. Though this seems problematic for formalisms like CCG, where there are infinite ways to compose two arbitrary elements to produce a third, this can usually be avoided by schematization. That is, instead of returning every possible result, the results can be summarized using variables.

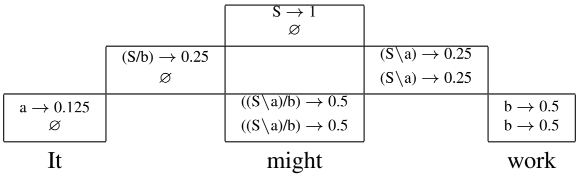


Figure 1: A chart showing the parse of the sentence *It might work*. Parse values are given at the bottom of each node, while learn values are given at the top.

2.2 The Chart

The Missing Link algorithm is built around a CKY-style chart. The chart consists of cells representing potential parses for each substring of the sentence. In Missing Link, each cell stores two separate analyses: one for the potential bottom-up parses of the sentence, and one for the potential top-down decompositions of the sentence, based on the category assigned to the sentence. These will be referred to as the "parse value" and the "learn value" for each cell.

Both the parse value and the learn value are represented using the same data structure, which is also used in the lexicon. This data structure maps potential categories (elements of E according to the target formalism) to probabilities, which represent the algorithm's confidence that the given category is valid for the corresponding substring. Note that the sum of the probabilities for the different categories is not necessarily 1: in the case where the algorithm is certain that a substring is ambiguous, the probabilities will sum to at least 2. Missing Link does not assign probabilities based on frequency relative to other substrings, sentences, or categories and makes no distinction between alternative parses other than their probability of being grammatical.

An example chart is shown in Figure 1.

2.3 Parsing

For each sentence in the training data, the Missing Link algorithm begins by looking up each word in the lexicon. The results of the lookup are assigned to the parse value for the bottom cells of the chart, which represent the length-1 substrings. The algorithm then attempts to parse as much of the sentence as possible. Initially, it will not be possible to parse any sentences, as no words have been introduced to the lexicon. However, as the algorithm sees more sentences in the training set, the lexicon will expand and the parses will become more complete.

The parsing step is fairly typical for a probabilistic CKY parser, proceeding in a bottom-up direction by calling COMPOSE for each pair of adjacent substrings. The only major difference does not come from the parsing strategy *per se*, but from necessary constraints on the formalisms compatible with Missing Link.

The confidence values for subsequent cells in the chart are determined under the assumption that all assignments are independent. Thus, in most cases, $P(\text{COMPOSE}(A, B)) = P(A)P(B)$. If the same category is found multiple times, these values are also treated as independent; thus, $P(A) = P(A_1) + P(A_2) - P(A_1)P(A_2)$, where A_1 and A_2 are separate instances of the category A.

Once as much of the chart has been filled by the parser as possible, the parse stage ends for that sentence. The parse values of the cells are retained when the chart is passed to the learning stage; incomplete parses are used to inform the learning stage. Even if the parse was successful, the learning stage still occurs, in order to update the system's confidence in the values used to produce the successful parse.

2.4 Learning

The learning stage is similar to the parsing stage, although it is somewhat more involved as it is able to take advantage of the results of the parse to minimize its search space.

First, the learn value of the root of the chart is initialized with the annotated category for the sentence. Then, the algorithm proceeds in a top-down manner calling DECOMPOSE on each cell and two corresponding substrings. For the most part, this proceeds in the same manner as the parsing stage, except in a top-down direction. There are, however, a few differences.

The base probability for a learned category is based on the probability of the root and the probability of any known sub-constituents.

When assigning probabilities in the learning stage, the learner must contend with the fact that there are multiple possibilities and no guarantee that they are all valid. The learner must also contend with the fact that there are multiple possible tree structures. Since chart parsing deals only with binary-branching trees, it is possible to calculate the number of possible trees for a sentence of a given length. The probability must then also be modulated by the $n - 1$st Catalan number, where n is the length of the substring corresponding to the current cell.

Thus, the total probability for a given result is equal to $\frac{p}{C_{n-1}l}$ where p is the base probability, C_n is the nth Catalan number, n is the length of the substring, and l is the number of results produced by DECOMPOSE.

The results are also dependent on the parse values of the corresponding sub-constituents. If both of the sub-constituents have known parse values, then learning does not necessarily need to occur. If these sub-constituents can be composed to produce at least one value in the current cell's learn value, then those sub-constituents will be added to the learn values of their cells, with their original probabilities. In other words, if the values are known and can produce the target value, then no additional learning needs to occur. If the target value cannot be produced, then learning occurs as normal, as if neither value were known.

A similar situation occurs when only one of the sub-constituents is known. In this case, DECOMPOSE is applied as normal, with the restriction that the known value remains constant. If DECOMPOSE is successful, then the probability remains constant as well. On the other hand, if DECOMPOSE cannot learn any values, then the decomposition is attempted again, as though neither value were known.

2.5 Lexical Update

Once both parsing and learning have occurred for a given sentence, the algorithm updates the lexicon based on the learned values for the length-1 substring cells in the chart. Since values in the lexicon are represented the same way as cells in the chart, this is a fairly straightforward process. If the category for a word in the current sentence is the same as a category already in the lexicon, the probability of the word is updated according to the assumption that the two probabilities are independent, as described above.

3 Implementing Combinatorial Categorial Grammar

For the purposes of this paper, Combinatory Categorial Grammar (CCG) will be used as an example formalism with Missing Link. CCG is an efficiently parseable grammar formalism in which all language-specific rules are stored in the lexicon (Steedman and Baldridge, 2002).

To implement a formalism for Missing Link, it is only necessary to define the set E, and the functions COMPOSE and DECOMPOSE. In CCG, the set E will be the set of categories, which is defined as follows:

- Given a set A of atoms, if $a \in A$, then a is a category.
- If a and b are categories, then (a\b) and (a/b) are categories. These are referred to as complex or functional categories.

For the purposes of Missing Link, it is also necessary to define one additional type of category: the variable. In this paper, variables are represented using lowercase letters while atoms are represented using capital letters.

In this paper, I start with the variant of CCG defined in Eisner (1996). In this variant, the COMPOSE operator can be defined fairly straightforwardly using two combinators:

$$\frac{(X/Y) \qquad X|_n \cdots |_2 Z_2 |_1 Z_1}{X|_n Z_n \cdots |_2 Z_2 |_1 Z_1} > Bn$$

$$\frac{X|_n \cdots |_2 Z_2 |_1 Z_1 \qquad (X\backslash Y)}{X|_n Z_n \cdots |_2 Z_2 |_1 Z_1} < Bn$$

These combinators, generalized forward composition and generalized backward composition, respectively, result in a TAG-equivalent formalism. I use Steedman's result-first notation for CCG categories and assume left-associativity unless disambiguated by parentheses.

The COMPOSE function for CCG can be defined as simply taking two categories and returning the results of either of these combinators. The exception is when one of the categories contains a variable; in this case, if the variable can be assigned to a value in the other category to produce a valid combination, then it will be. For more details on variables, see Section 3.2.

The DECOMPOSE, as the inverse of COMPOSE, can be derived from the same combinators. In effect, the algorithm attempts to match as much of the pattern as is available in order to construct the rest of the proof. Any values which cannot be determined concretely are replaced with variables. For example, given the root X, this matches both combinators where $n = 0$, allowing the function to select ((X/a), a) and (b, (X\b)) as potential results.

3.1 Modalities

As written above, this formalism is performs poorly with Missing Link. Though COMPOSE and DECOMPOSE always produce finite results, the generalized combinators prove problematic for learning. This is because crossed composition in CCG can result in permutation, which makes it impossible for Missing Link to distinguish between certain alternatives involving functional categories.

For example, consider the first learning instance involving a two-word sentence with no words known. This will produce a derivation such as the following, where multiple categories on a node correspond to alternative hypotheses proposed by Missing Link.

$$\begin{array}{ccc} & \text{S} & \\ / & & \backslash \\ \text{(S/a)} & & \text{a} \\ \text{b} & & (\text{S}\backslash\text{b}) \end{array}$$

If the algorithm then attempts to parse this same sentence (or any similar one), it will run into a problem. Since the algorithm cannot tell which hypothesis for the left node was originally paired with which hypothesis for the right node, it is forced to try all possibilities during composition. This will result in the algorithm attempting to compose (S/a) and (S\b). If $a = S$, then this can compose according to the generalized composition rules above, resulting in (S\b). However, the algorithm has made a crucial mistake in assigning S to the input type b. In fact, because these situations crop up whenever an atom or variable is decomposed, this sort of assignment is exceedingly common. In addition, because S is usually the only concrete atomic category (other atoms must be represented by clusters of variables, since Missing Link is unsupervised!), eventually *all* variables tend to converge to S. This results in an incomprehensible grammar in which the only atom is S.

One possible solution would be to prevent hypotheses from composing with competing alternatives (i.e., alternatives generated at the same time). However, this solution would not be sufficient, as the same situation could still occur with other words – the pattern that created the first set of hypotheses will apply to other words as well. Thus, convergence to S will still occur in these situations.

The solution that I used was to take advantage of the slash modalities used in many CCG variants, such as Baldridge (2002). These variants place modalities on the slash categories that restrict their application to certain combinators. I use four basic modalities, taken from Steedman and Baldridge (2002).

- Star ($*$): Categories with this slash can only apply in simple applicative contexts (function composition of all types is forbidden).
- Diamond ($\diamond$): Categories with this slash can compose only with categories of the same slash direction.
- Cross ($\times$): Categories with this slash can compose only with categories of the opposite slash direction.
- Dot ($\cdot$): Categories with this slash can compose with any other category.

Using these modalities restricts the cases where certain compositions may occur. During learning, the most restrictive category that applies to the given decomposition is always used. For example, the decomposition of S results in ((S/$*$a), a) and (b, (S$\backslash *$b)). Due to the restrictions on the modalities, it is no longer possible to compose the results using crossed composition: (S/$*$a) and (S$\backslash *$b) are incompatible! The problem of convergence to S no longer exists, and the computational properties of the formalism are maintained: crossed composition can still occur as a last resort, in cases where it is clear that it can be derived.

As an aside, it is no accident that crossed composition was the cause of this issue. The permutations caused by crossed composition also make it necessary in mildly context-sensitive CCGs to account for data in languages such as Swiss German and Dutch. It is interesting, though not surprising, that this comes with its own difficulties in learning.

3.2 Variables and State

As described in previous sections, this implementation of CCG for Missing Link uses variables to represent categories whose exact value cannot be determined. Though variables are necessary, they also introduce additional complexity into the algorithm. There are a few special notes that relate to the treatment of variables in CCG.

The values of variables are stored in a global state, which maps variable IDs to their values (if a value is known). When a variable is evaluated in parsing or learning, it is first resolved according to the state. In most cases, a variable cannot be resolved completely (the final representation will still contain one or more variables); this is expected, as the algorithm is not able to induce new atomic categories, it must instead make use of variables that are designated to represent new categories.

The value of a variable may be a complex category, an atomic category, or another variable (the latter case being used primarily to set two variables equal to one another). If a variable is set equal to a complex category, it may be that the complex category itself contains variables. This creates the possibility of reference cycles, which would produce undefined values. To avoid this, every time an assignment is made, the algorithm performs an occurs check to ensure that the assignment will not produce any reference cycles: the new value is checked for any instances, direct or indirect, of the variable. If there are any, the assignment cannot be completed and the algorithm must try another alternative.

Once a variable is assigned a value, it is permanent. However, the algorithm is still free to introduce a new variable by re-decomposing categories in future training samples, according to the rules given in Section 2. Furthermore, if a variable assignment fails within a combinator (due to an occurs check), the state is rolled back to the way it was before the combinator began processing. This prevents known inconsistencies from filling the state; though the state may still contain inconsistencies, all values in the state are part of a potential analysis. Any remaining inconsistencies typically do not achieve high probability during learning, as they cannot be used to successfully parse many sentences.

4 Evaluation

Evaluating an unsupervised learning algorithm is a difficult prospect. Though many unsupervised syntactic learning algorithms are evaluated by comparing the resulting dependency structures to a gold standard, typically by ensuring that each predicted dependency is directed in the same way between the same two lexical units as the gold standard dependency. However, comparison by dependency structures is not always a good choice for unsupervised learning algorithms. In particular, because one of the goals of Missing Link is to provide a framework for analyzing grammar formalisms in an otherwise theory-independent manner, it is undesirable to make use of any theory-dependent analysis. Though dependencies may be largely independent of theory, they still make conventional decisions. For example, the Universal Dependencies project does not usually allow functional categories to be heads, while alternatives, such as Stanford Dependencies, do allow functional heads. If the learning algorithm is free to choose from the alternatives on its own, then it cannot be accurately evaluated against such a standard.

To evaluate Missing Link, I therefore use a secondary task. Similar to BERT (Devlin et al., 2018), I use Missing Link as an unsupervised pre-training algorithm for a downstream task. In this case, I use linguistic acceptability (grammaticality) judgments as the downstream task. This is an ideal task for Missing Link, since Missing Link's confidence values essentially capture the notion of probability that a sentence (or substring) is grammatical. I use the Corpus of Linguistic Acceptability (Warstadt et al., 2018) to provide the gold standard and training data. The Corpus of Linguistic Acceptability (CoLA) provides a basic classification task in which sentences are annotated with boolean grammaticality judgments – that is, each sentence is either considered grammatical or not. Missing Link will provide the pre-trained linguistic model, and a simple logistic regression will use Missing Link confidence values to classify the test data.

4.1 Pre-Training

Before training the model, Missing Link is used to pre-train a linguistic model of the target language, in this case English. In order to keep the conditions between the training set and the testing set as close as possible, it is necessary to pre-train Missing Link on a different dataset than the annotated linguistic acceptability data. In addition, the corpus of linguistic acceptability is relatively small. To this end, I pre-train Missing Link using the much larger Billion Words corpus (Chelba et al., 2013).

For pre-training, I sorted the sentences of the Billion Words corpus (BWB) in ascending order by length. Missing Link is able to assign higher confidence to words in shorter sentences. This allows it to have a relatively small set of hypotheses for many words that occur in shorter sentences, which it can then use to better learn nearby words in longer sentences. Sentences of length less than 3 were excluded, since most short sentences in BWB are simple noun phrases or noisy punctuation, which can confuse Missing Link. Sentences of length greater than 10 were excluded as well, since they tend to contribute little to Missing Link's confidence.

For practical reasons, I also restricted the number of categories learned for each cell in the chart and lexical entry to 50, to improve efficiency while still allowing for multiple simultaneous hypotheses.

4.2 Training and Testing

Once the linguistic model is pre-trained, then it can be used to train a logistic regression. To train the logistic regression, Missing Link processes each sentence in the CoLA training set, producing a confidence value for each potential sentential category. If no parse is produced, or if S is not in the results, then Missing Link is allowed to learn from the sentence. By learning from sentences where no valid parse was produced, Missing Link becomes robust to sentences where some words were not in the original pre-training set. After learning, Missing Link attempts to parse the sentence again.

The confidence of the category S is used as one independent variable for training the logistic regression. The other independent variables are the length of the sentence, whether the category S was found (including after re-training), and whether retraining was required. Longer sentences are inherently associated with lower probabilities due to the independence assumptions used in composition. Sentences where the category S was never found are also distinguished from sentences where the probability of S was negligible; although both are likely to indicate an ungrammatical sentence, for long sentences, the distinction may be necessary. Lastly, whether retraining was necessary is a plausible predictor as well, since retraining indicates that either some words are out of vocabulary or the sentence could not be parsed with the previously learned categories.

Once the logistic regression has been fit to the training data, the same process is applied to the testing data in order to predict whether each sen-

Model	MCC
MT-DNN	68.4
RoBERTa	67.8
XLNet-Large	67.8
GLUE Human Baseline	66.4
Missing Link	**63.0**
XLM	62.9
BERT-24	60.5

Table 1: CoLA Benchmark[2]

tence is grammatical or not.

5 Results and Conclusion

The Corpus of Linguistic Acceptability is evaluated using Matthews Correlation Coefficients, since there are far more grammatical sentences in the data than ungrammatical ones. A number of other systems have been tested against CoLA and can be used as benchmarks for Missing Link, since the same training-testing split is used by all systems.

The results of the evaluation of Missing Link as well as some of the top performing competitors are given in Table 1. With an MCC of 63.0, Missing Link does not advance the state-of-the-art compared to deep learning models, but it does perform competitively. Given that Missing Link uses only a basic logistic regression on top of the pre-trained model, this presents evidence that Missing Link is producing a reasonable grammar for the data.

Given that Missing Link produces a reasonable grammar, it can then be used for further study in the fields of grammar formalisms and theoretical linguistics. Different grammar formalisms can be compared using the same core algorithm, allowing for any variation in performance to be attributed to properties of the grammar formalism. The algorithm can also be used to explore specific linguistic phenomena from a learning perspective. Given two alternatives to a linguistic phenomenon, it is possible to use Missing Link as one potential way of distinguishing between the two. This presents a new paradigm in linguistic research as a means of exploring generative linguistics through a formal, but nuanced, model of learning and learnability. Missing Link is not necessarily the only algorithm that supports this paradigm, but presents evidence that such a paradigm is feasible for linguistic theory.

[2]These baselines are taken from the GLUE leaderboard at the time of writing (Wang et al., 2019).

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Jason Baldridge. 2002. *Lexically specified control in combinatory categorial grammar*. Ph.d. dissertation, University of Edinburgh.

Yonatan Bisk, Christos Christodoulopoulos, and Julia Hockenmaier. 2015. Labeled grammar induction with minimal supervision. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 870–876, Beijing, China. Association for Computational Linguistics.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorston Brants, and Phillip Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

Jinho D. Choi and Andrew McCallum. 2013. Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, pages 1052–1062.

Jinho D. Choi, Joel Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 387–396.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Shimon Edelman, Zach Solan, David Horn, and Eytan Ruppin. 2003. Rich syntax from a raw corpus: Unsupervised does it. In *Syntax, Semantics and Statstics Workshop at NIPS '03*, Whistler, British Columbia, Canada.

Jason Eisner. 1996. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 79–86. Santa Cruz.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. Unpublished manuscript.

William P. Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing.

In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 101–109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aravind K. Joshi, K. Vijay Shankar, and David Weir. 1990. The convergence of mildly context-sensitive grammar formalisms. Technical Report MS-CIS-90-01, Department of Computer and Information Science, University of Pennsylvania.

Tadao Kasami. 1965. An efficient recognition and syntax-analysis algorithm for context-free languages. Technical report, AFCRL.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1512–1523, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karne Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the workshops on Human Language Technology*, pages 114–119. Association for Computational Linguistics.

André F.T. Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the ACL*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zema. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of LREC*, pages 1659–1666.

Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. 2011. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1281–1290, Stroudsburg, PA, USA. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "Less is More" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 751–759, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Steedman and Jason Baldridge. 2002. Combinatory categorial grammar. In Robert D. Borsley and Kersti Börjars, editors, *Non-Transformational Syntax*, pages 181–224. Blackwell.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *CoRR*, abs/1805.12471.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189–208.

Assessing the ability of Transformer-based Neural Models to represent structurally unbounded dependencies

Jillian K. Da Costa
School of Management
University at Buffalo, SUNY
jillian.k.dacosta@buffalo.edu

Rui P. Chaves
Department of Linguistics
University at Buffalo, SUNY
rchaves@buffalo.edu

Abstract

Filler-gap dependencies are among the most challenging syntactic constructions for computational models at large. Recently, Wilcox et al. (2018) and Wilcox et al. (2019b) provide some evidence suggesting that large-scale general-purpose LSTM RNNs have learned such long-distance filler-gap dependencies. In the present work we provide evidence that such models learn filler-gap dependencies only very imperfectly, despite being trained on massive amounts of data. Finally, we compare the LSTM RNN models with more modern state-of-the-art Transformer models, and find that these have poor-to-mixed degrees of success, despite their sheer size and low perplexity.

1 Introduction

A flurry of recent work has shown that modern large-scale and general-purpose Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) achieve impressive results as computational psycholinguistic models of human language processing, such as Linzen et al. (2016), Gulordava et al. (2018), Linzen and Leonard (2018), van Schijndel and Linzen (2018), Futrell et al. (2018), and Wilcox et al. (2019a)), to list only a few. Some of this work has focused on long-distance dependencies like (1), involving a linkage between a *wh*-phrase and a gap. This is one of the phenomena that Markovian language models have always been inherently bad at.

(1) I know **who**$_i$ the gardener reported the butler said the hostess believed her aunt suspected you delivered a challenge **to** $\_{}_i$ at the party. (Wilcox et al., 2019b)

However, such long-distance dependencies are accompanied by morphosyntactic constraints which have not previously been tested, in particular, agreement constraints like those in (2).

(2) a. It was the lawyer who I think you said _ was/*were upset.
   b. It was the lawyers who I think you said _ *was/were upset.
   c. They wondered which lawyer I think you said _ was/*were upset.
   d. They wondered which lawyers I think you said _ *was/were upset.

There are two different dependencies at work in the these examples. One is between the filler phrase *who* and the gap (i.e. the missing subject of the embedded verb) and another between the head noun *lawyer*(*s*) and the *wh*-phrase adjacent to it. It is not possible to claim that LSTM RNN models have learned English filler-gap dependencies without showing that the associated morphosyntactic constraints have also been learned. At the time of this writing, LSTM RNNs are no longer the state-of-the-art English language models. Transformer (attention-based) models have obtained lower test-time perplexity. In the present work we focus on whether any of these neural language models have truly learned long-distance agreement (filler-gap) dependencies like those in (1) and (2).

The structure of the paper is as follows. First we show that the same general-purpose LSTM RNN models that Wilcox et al. (2019b) have claimed to successfully cope with filler-gap dependencies have not learned the morphosyntactic constraints associated to such constructions, illustrated in (2). Next, we compare these results with those of three more recent transformer-based architectures that have obtained better perplexity results, namely Transformer-XL (Dai et al., 2019), BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), and OpenAI GPT-2 (Radford et al., 2019).[1]

[1] All our materials, code, and analysis are available at https://github.com/RuiPChaves/Transformers-FillerGap-dependencies.

We acknowledge that these models are not directly comparable, and that the present results should be taken with some caution because the architectures are different (transformer vs. recurrent), as are the training objectives (masked language modeling vs. non-masked language modeling), evaluation methods (use of sentences prefix + suffix vs. only prefix for language models), and the training datasets. Nonetheless, we argue that such a preliminary comparison is useful in that is sheds some light on how well extremely large neural models of English cope with perhaps of the most historically vexing syntactic phenomena in computational linguistics. As we shall see, there is a wide range of variation in how accurately the models cope with filler-gap dependencies, with LSTM RNNs fairing among the worse. Our results are consistent with those reported by Goldberg (2019), which suggest that BERT is better than LSTM RNNs at English subject-verb agreement (Marvin and Linzen, 2018).

2 LSTM RNNs

Wilcox et al. (2019b) found evidence suggesting that LSTM RNNs can maintain filler-gap dependencies across up to at least four clausal boundaries like the one in (1). Two models were used for these experiments. One was Gulordava et al. (2018), henceforth the **Gulordava model**, which was trained on 90 million tokens of English Wikipedia, and has two hidden layers of 650 units each. The second model was Jozefowicz et al. (2016), henceforth the **Google model**, which was trained on the One Billion Word Benchmark (Chelba et al., 2013), has two hidden layers with 8196 units each, and uses the output of a character-level Convolutional Neural Network as input to the LSTM. One of the trademark properties of filler-gap dependencies is that the morphosyntactic properties imposed on the gap site are preserved by the filler phrase, as already illustrated in (2). Here, the plural noun must be matched with the plural verb form and the singular noun with the singular verb. In what follows we examine how well these dependencies are learned by the Gulordava and Google models.

2.1 Experiment 1: agreement in clefts

Following basically the same experimental approach as Wilcox et al. (2018), we created 20 cleft items using a $2\times2\times4$ factorial design, for a total of 320 sentences. All the conditions for an item are illustrated in (3). Like Wilcox *et al.*, we extracted the softmax activation of the critical verbs *were*/*was*, given the prefix sentence, using basically the same code as Wilcox et al. (2018), made available at `https://osf.io/zpfxm/`.

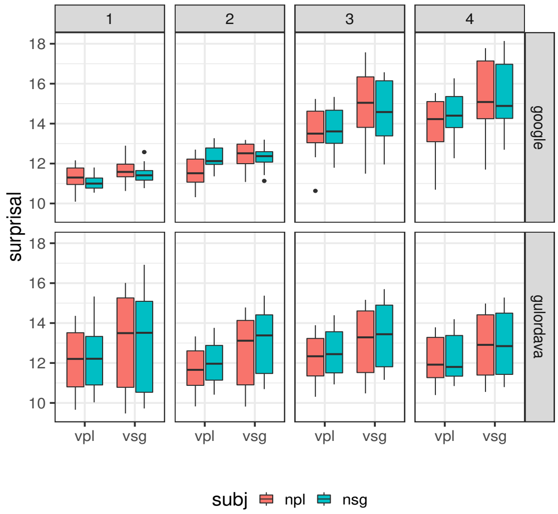


Figure 1: Surprisal of the gap-agreeing verb in 'it' clefts across 4 levels of embedding (LSTM RNNs)

(3) a. It was the lawyer(s) who I think was/were ... [$N_{sg/pl}$, LEVEL1, $V_{sg/pl}$]

b. It was the lawyer(s) who I think you said was/were ... [$N_{sg/pl}$, LEVEL2, $V_{sg/pl}$]

c. It was the lawyer(s) who I think you said you thought was/were ... [$N_{sg/pl}$, LEVEL3, $V_{sg/pl}$]

d. It was the lawyer(s) who people believe I think you said you thought was/were ... [$N_{sg/pl}$, LEVEL4, $V_{sg/pl}$]

Finally, we converted the softmax values into surprisal (i.e. the negative log probability), following Wilcox et al. (2019b). See see Hale (2001) and Levy (2008) for more discussion.

The results were rather weak, as shown by Figure 1. Had the RNNs succeeded at this task, then the conditions where the noun and verb agree (i.e. N_{pl}-V_{pl} and N_{sg}-V_{sg}) would be lower in surprisal than the conditions where the agreement is mismatched (i.e. N_{pl}-V_{sg} and N_{sg}-V_{pl}). This was generally not the case in either model. Finally, in the larger Google model surprisal increased with the level of embedding, so that the correct verb form is more unexpected in level 4 than the incorrect verb forms in levels 1 and 2.

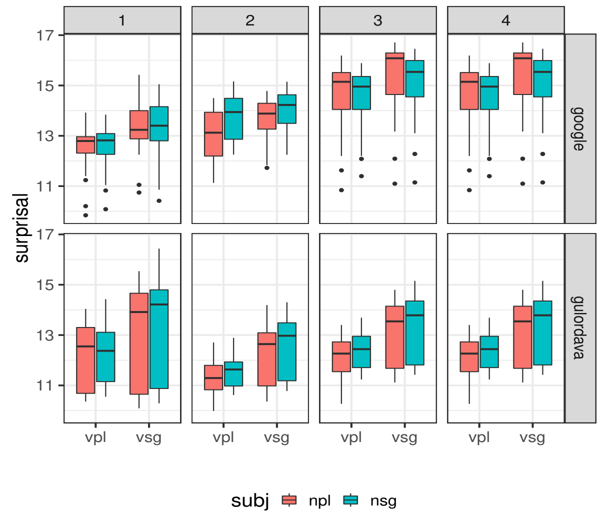


Figure 2: Surprisal of the gap-agreeing verb in 'which' interrogatives across embedding levels (LSTM RNNs)

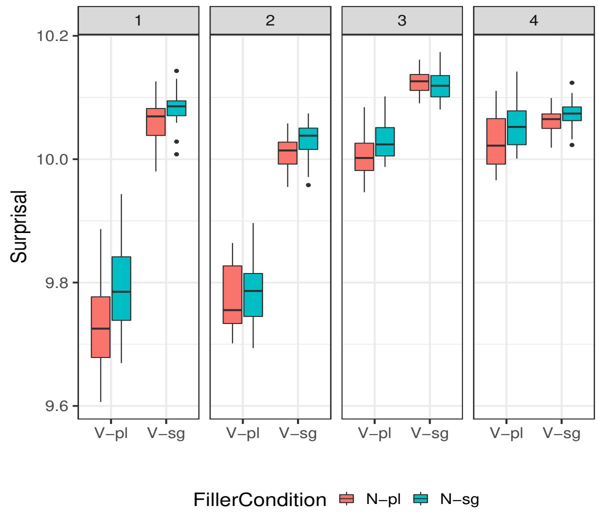


Figure 3: Surprisal of the gap-agreeing verb in 'it' clefts across embedding levels (Transformer-XL)

There is a general increase of surprisal as clausal embedding increases, which in our view may simply reflect the fact that multiple occurrences of embedded declarative clauses under verbs of indirect discourse are rare. Overall, the results suggest that these models have not learned the morphosyntax of filler-gap dependencies.

2.2 Experiment 2: agreement in indirect interrogatives

In order to assess if these results are specific to the cleft construction, we converted the 20 items into embedded interrogatives, effectively inverting the order of the *wh*-phrase and the agreeing nominal head, as (4) illustrates.

(4) a. Someone wondered which lawyer(s) I think was/were ...
[$\mathrm{N}_{sg/pl}$, LEVEL1, $\mathrm{V}_{sg/pl}$]

b. Someone wondered which lawyer(s) I think you said was/were ...
[$\mathrm{N}_{sg/pl}$, LEVEL2, $\mathrm{V}_{sg/pl}$]

c. Someone wondered which lawyer(s) I think you said you thought was/were ...
[$\mathrm{N}_{sg/pl}$, LEVEL3, $\mathrm{V}_{sg/pl}$]

d. Someone wondered which lawyer(s) who people believe I think you said you thought was/were ...
[$\mathrm{N}_{sg/pl}$, LEVEL4, $\mathrm{V}_{sg/pl}$]

The outcome was the same, as Figure 2 illustrates, suggesting that our results are robust and not specific to the type of filler-gap construction chosen. We conclude that the Gulordava and Google models have not truly learned the morphosyntax of filler-gap dependencies. In what follows we examine how more recent transformer-based models fair at the same tasks.

3 Transformer-XL

Transformer-XL (Dai et al., 2019) has 24 million parameters, an average attention span of 640 tokens, and 16 10-word transformer layers. Transformer-XL is supposed to learn dependencies that are about 80% longer than those learned by RNNs but as Figure 3 shows, it did only marginally better than the Google and the Gulordava models when processing the same agreement in clefts dataset from Experiment 1.

In fact, only in embedding level 1 was the surprisal of agreeing N-V pairs statistically lower than their non-agreeing counterparts (for N_{pl}-Vpl vs. N_{sg}-Vpl we have t = -2.39, p = 0.021, and for N_{sg}-Vsg vs. N_{pl}-Vsg we have t = -1.83, p = 0.068). For all other levels of embedding there was no statistical difference in surprisal ($p > 0.4$), except for level 3 where N_{pl}-Vpl vs. N_{sg}-Vpl (t = -2.13, p = 0.039). The model does equally bad on the indirect interrogatives dataset from Experiment 2, as Figure 4 illustrates.

3.1 Experiment 3: Filler-gap surprisal in subject-inverted interrogatives

For completeness, we also tested Transformer-XL's ability to maintain a filler-gap dependency without the interacting factor of subject-verb

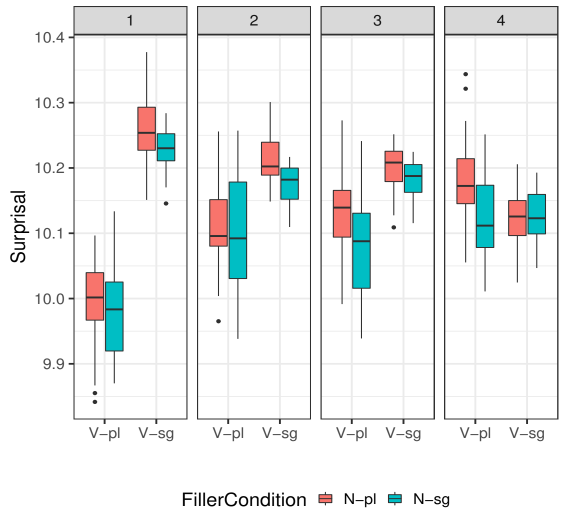


Figure 4: Surprisal of gap-agreeing verb in 'which' interrogatives across embedding levels (Transformer-XL)

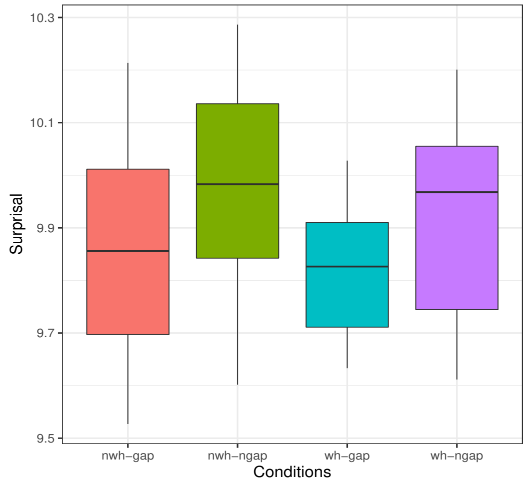


Figure 5: Surprisal of the post-gap region in inverted interrogatives at embedding level 1 (Transformer-XL)

agreement. We created 20 items, in a $2 \times 2 \times 4$ design, for a total of 320 sentences, as illustrated in (5). We extracted the softmax value of the masked post-gap region item (below, the preposition *at*). This experiment serves as the counterpart of the experiments in Wilcox et al. (2019b) showing LSTM RNNs can maintain filler-gap dependencies across up to at least four clausal boundaries (diacritic '*' not included in the input).

(5) a.*What did we talk about it at the party? [WH-NOGAP, LEVEL1]

b. What did we talk about _ at the party? [WH-GAP, LEVEL1]

c. Did we talk about it at the party? [NOWH-NOGAP, LEVEL1]

d.*Did we talk about _ at the party? [NOWH-GAP, LEVEL1]

The results confirm that Transformer-XL has a poor representation for filler gap dependencies, as seen in Figure 5. Already at one level of embedding the surprisal of the (ungrammatical) nowh-gap condition is lower than the grammatical nh-gap counterpart, whereas it should be the other way around. In levels 2 through 4 there is no statistical difference between any of the four conditions.

3.2 Experiment 4: Filler-gap surprisal in uninverted indirect interrogatives

In order to determine if the results of Experiment 3 scale to other filler-gap constructions, we constructed non-inversion counterparts of the 20 items, illustrated in (6). As before, we extracted the softmax activation of the critical verbs at the end of the item, after the sentence prefix is processed. The results were similar in that in no level of embedding the correct surprisal pattern was observed. See the materials for details.

(6) a. People wondered what we talked about it at ... [WH-NOGAP, LEVEL1]

b. People wondered what we talked about _ at ... [WH-GAP, LEVEL1]

c. People wondered if we talked about it at ... [NOWH-NOGAP, LEVEL1]

d. People wondered if we talked about _ at ... [NOWH-GAP, LEVEL1]

We conclude that the English Transformer-XL model does much worse than the English LSTM RNNs in coping with filler-gap dependencies.

4 BERT

Google's Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based model that learns bidirectional encoder word representations via a masked language model training objective, using 340 million parameters, 768 hidden layers, 24 transformer blocks, and 1020 word context windows.

Using the same agreement in filler-gap dependencies dataset used in Experiment 1, we probe whether BERT assigns relative probability to plural and singular verb forms in such a way that this consistent with the agreement information of the

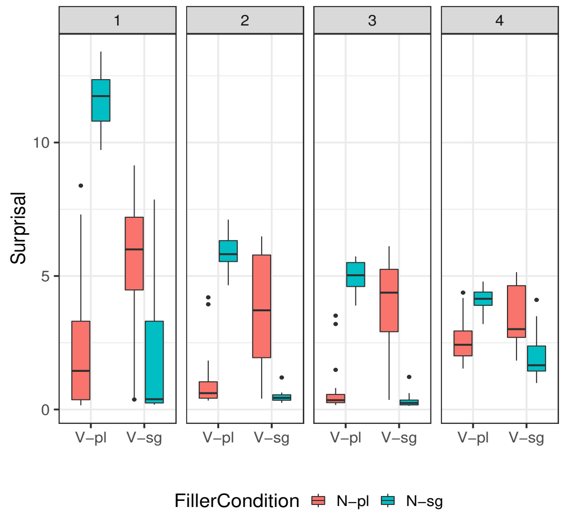


Figure 6: Surprisal of the gap-agreeing verb in 'it' clefts across 4 levels of embedding (BERT)

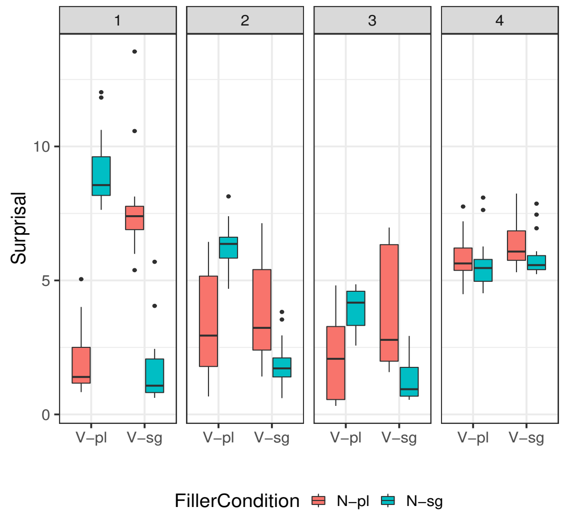


Figure 7: Surprisal of the gap-agreeing verb in 'which' questions across embedding levels (BERT)

nominal antecedent at the top of the filler-gap dependency. Like Goldberg (2019) and Wolf (2019), we masked the verb and then extracted the softmax values for both *was* and *were*, as shown in (7).

(7) a. It was the lawyer(s) who I think [MASK] upset. [$N_{sg/pl}$, LEVEL1]

b. It was the lawyer(s) who I think you said [MASK] upset. [$N_{sg/pl}$, LEVEL2]

c. It was the lawyer(s) who I think you said you thought [MASK] upset. [$N_{sg/pl}$, LEVEL3]

d. It was the lawyer(s) who people believe I think you said you thought [MASK] upset. [$N_{sg/pl}$, LEVEL4]

The results are much better than those obtained by LSTM RNNs on the same items, as Figure 6 illustrates. The surprisal of the agreeing conditions is systematically lower than that of the non-agreeing conditions in all embeddings (all $ps < 0.0001$).

In the next experiment, the 20 items were converted the *which* interrogative counterparts, analogously to Experiment 2 above, where the agreeing verb were masked, as seen in (8).

(8) a. Someone wondered which lawyer(s) I think [MASK] upset. [$N_{sg/pl}$, LEVEL1]

b. Someone wondered which lawyer(s) I think you said [MASK] upset. [$N_{sg/pl}$, LEVEL2]

c. Someone wondered which lawyer(s) I think you said you thought [MASK] upset. [$N_{sg/pl}$, LEVEL3]

d. Someone wondered which lawyer(s) who people believe I think you said you thought [MASK] upset. [$N_{sg/pl}$, LEVEL4]

The results are in Figure 7, and are only weak in embedding level 4, where neither condition is statistically different in the V-sg (t = 0.91, $p = 0.36$) nor in the V-pl (t = 1.93, $p = 0.06$) conditions.

If BERT's ability to maintain filler-gap dependencies in memory is too superficial and eager, then it may ignore the presence of a local subject, and not recognize that a subject gap is grammatically impossible, as in (9).

(9) a.*It was the boys who I think she/he were lost [N_{pl}, V_{pl}, LEVEL1]

b.*It was the boy who I think we/they was lost. [N_{sg}, V_{sg}, LEVEL1]

For example, if the model attempts to link *boys* to the copula verb in (9a) despite the local subject pronoun, then the surprisal of *were* should be higher than that of *was*. Similarly, if the model attempts to link *boy* to the copula verb in (9b) despite the local subject pronoun, then the surprisal of *was* should be lower than that of *were*. The presence of the pronoun makes the subject gap impossible, and BERT should be sensisitive to that.

We therefore inserted a pronoun in the gap site of the 20 items used in the experiment immediately above, and made sure the verb agreed with the fronted phrase, not the pronoun. What we found was a complete reversal of the surprisal values. As Figure 8 shows, BERT suspends the filler-

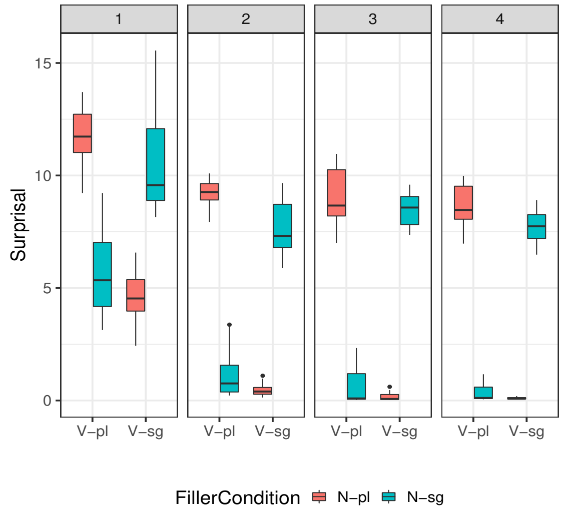


Figure 8: Surprisal of the (dis)agreeing verb in 'it' clefts across 4 levels of embedding (BERT)

gap linkages in the copula of examples like (9). We conclude that BERTs processing of filler-gap dependencies is not trivially shallow.

As in Experiments 3 and 4 above, we also examined BERT's ability to maintain a filler-gap dependency without the interacting factor of subject-verb agreement. Using the same items as in §3.1 and §3.2, illustrated in (10), we extracted the softmax value of the masked post-gap region item (below, the preposition *at*).

(10) a.*What did we talk about it at the party? [WH-NOGAP, LEVEL1]

b. What did we talk about _ at the party? [WH-GAP, LEVEL1]

c. Did we talk about it at the party? [NOWH-NOGAP, LEVEL1]

d.*Did we talk about _ at the party? [NOWH-GAP, LEVEL1]

As Figure 9 shows, BERT is able to represent the filler gap dependency up to four levels of clausal embedding. Surprisal is highest when there is a gap but no *wh*-phrase, and lower when (i) there is no gap and no *wh*-phrase and (ii) when there is a gap and a *wh*-phrase. The low surprisal obtained for the case where there is no gap and *wh*-phrase is more difficult to interpret, since the model's input has access to information about clausal boundaries. In that sense, the surprisal is lower than one would expect.

BERT faired equally well with the uninverted indirect interrogative counterparts of (5), shown in (11), which were identical to the items used in Experiment 4 above; see §3.2.

(11) a.*People wondered what we talked about it at the party. [WH-NOGAP, LEVEL1]

b. People wondered what we talked about _ at the party. [WH-GAP, LEVEL1]

c. People wondered if we talked about it at the party. [NOWH-NOGAP, LEVEL1]

d.*People wondered if we talked about _ at the party. [NOWH-GAP, LEVEL1]

BERT's masked language objective has an advantage over RNN models in that it has access to input after the masked critical item, e.g. the string *the party* in (5). We therefore ran a $2 \times 2 \times 4$ variant of Experiment 6 in which the masked critical items were adverbs like *yesterday*, *repeatedly*, *again*, and *then*, in sentence-final position:

(12) a.* What did we talk about it yesterday? [WH-NOGAP, LEVEL1]

b. What did we talk about _ yesterday? [WH-GAP, LEVEL1]

c. Did we talk about it yesterday? [NOWH-NOGAP, LEVEL1]

d.* Did we talk about _ yesterday? [NOWH-GAP, LEVEL1]

The results were radically different, as the surprisal was essentially inverted as shown in Figure 10. This pattern remained the same in deeper embedding levels, suggesting that BERT's ability to maintain filler-gap dependencies is brittle.

Finally, we also ran a variant of this experiment where the 20 items were converted into embedded interrogatives, without inversion. Again, the masked critical items were the adverbs in sentence-final position:

(13) a.* People wondered what we talked about it repeatedly. [WH-NOGAP, LEVEL 1]

b. People wondered what we talked about _ repeatedly. [WH-GAP, LEVEL 1]

c. People wondered if we talked about it repeatedly. [NOWH-NOGAP, LEVEL 1]

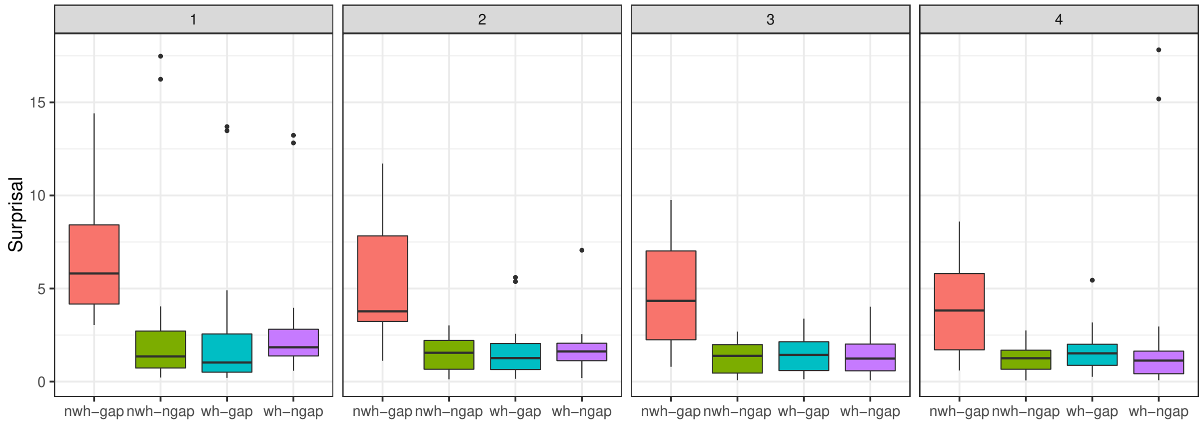


Figure 9: Surprisal of the post-gap region in subject-inversion interrogatives across embedding levels (BERT)

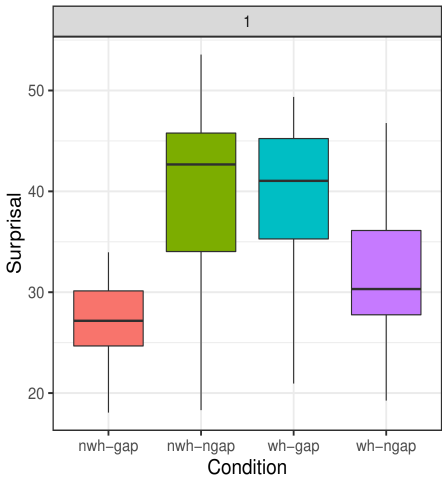


Figure 10: Surprisal of the sentence-final adverb in subject-inversion interrogatives, embedding 1 (BERT)

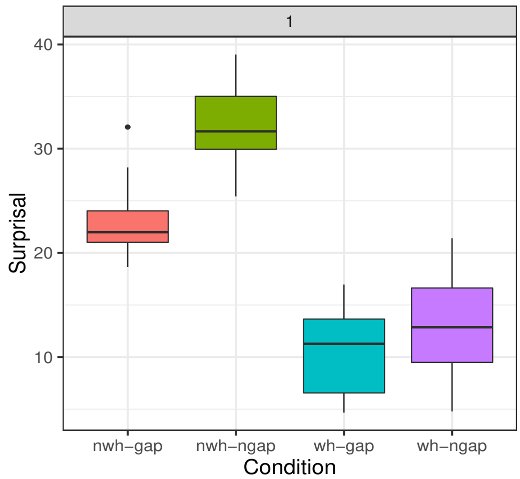


Figure 11: Surprisal of the sentence-final adverb in uninverted indirect interrogatives at embedding 1 (BERT)

d.* People wondered if we talked about _ repeatedly.
[NWH-GAP, LEVEL 1]

Now, the condition with the highest surprisal was nwh-ngap, suggesting that the model does not expect sentence-final adverbs to follow pronouns in the absence of a filler-gap dependency. The first level of embedding is shown in Figure 11. BERT's modelling of filler-gap dependencies is better than all other models surveyed so far but still brittle.

5 XLNet

XLNet (Yang et al., 2019) is like BERT in that it uses a masked model training objective and learns bidirectional contexts. Although XLNet is claimed to achieve better results than BERT in a number of tasks, we found that it performed worse in the same experiments ran on BERT, failing to provide clear evidence that filler-gap dependencies are attended to. For example, XLNet did much worse with clefts items, like those illustrated in (8). As can be seen in Figure 12, there is a significant overlap across subject-verb agreeing and non-agreeing conditions. Had the model learned about agreement in filler-gap dependencies, the surprisal of V-pl (*were*) in the N-pl condition should be significantly lower than that of V-pl in the N-sg condition. Similarly, the surprisal of V-sg in the N-pl condition should be significantly higher than that of V-sg (*was*) in the N-sg condition.

Similarly poor results were found for the interrogative subject-agreement items, like those in (7), as Figure 13 indicates. As in the case of Transformer-XL, there is little evidence that the model attends to filler-gap dependencies at all.

6 GPT-2

Unlike Google's BERT, the OpenAI GPT-2 model uses the same training objective as LSTM RNNs.

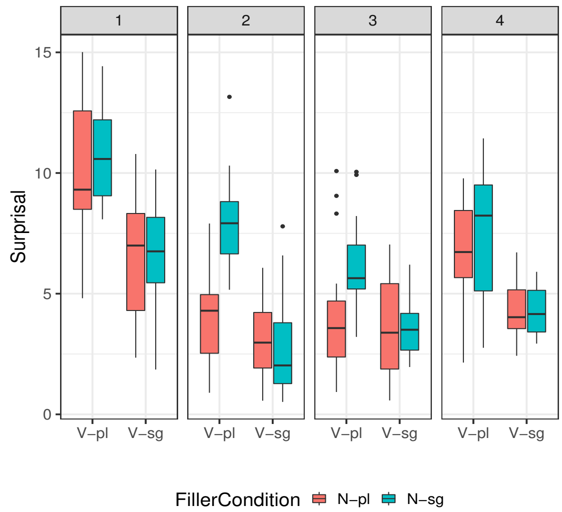


Figure 12: Surprisal of the gap-agreeing verb in 'it' clefts across 4 levels of embedding (XL-Net)

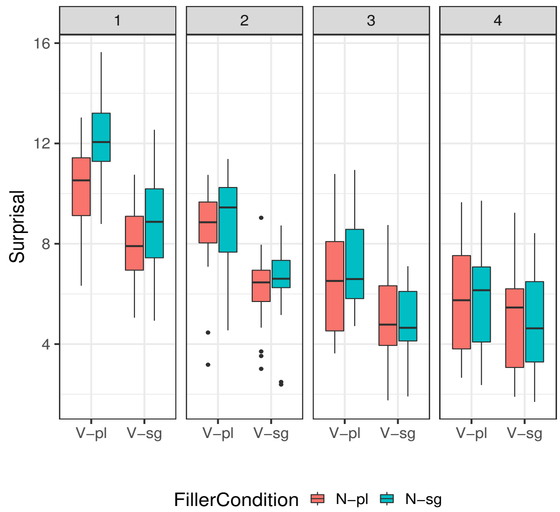


Figure 13: Surprisal of the gap-agreeing verb in 'which' questions across embedding levels (XLNet)

It is therefore possible to simply take the softmax activation of the word of interest after the sentence is processed. Preliminary evaluations on subject-verb agreement data by Wolf (2019) indicate that an earlier version of GPT-2 was worse than BERT on the Linzen et al. (2016) dataset but better in the more complex Marvin and Linzen (2018) dataset. In what follows, we report our findings for the more recent 345 million parameter version of GPT-2, hf. **GPT-2 medium**.

We begin with the 20 cleft items from Experiment 1, illustrated in (3), and repeated in (14). As before, we extracted the softmax activation of the words *was* and *were* across all conditions and converted the values to surprisal.

(14) a. It was the lawyer(s) who I think was/were ... [$N_{sg/pl}$, LEVEL1, $V_{sg/pl}$]

b. It was the lawyer(s) who I think you said was/were ... [$N_{sg/pl}$, LEVEL2, $V_{sg/pl}$]

c. It was the lawyer(s) who I think you said you thought was/were ... [$N_{sg/pl}$, LEVEL3, $V_{sg/pl}$]

d. It was the lawyer(s) who people believe I think you said you thought was/were ... [$N_{sg/pl}$, LEVEL4, $V_{sg/pl}$]

The GPT-2 medium results are shown in Figure 14, and are clearly superior to BERT's. For all levels of embedding, the agreeing conditions received statistically lower surprisal than that of the non-agreeing conditions. Notice how the differential across the conditions tends to diminish with

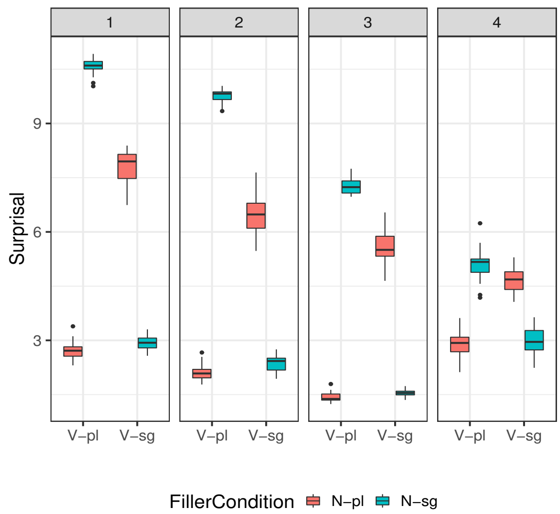


Figure 14: Surprisal of the gap-agreeing verb in 'it' clefts across 4 levels of embedding (GPT-2)

deeper clausal embeddings, suggesting that the dependency is lost in deeper embeddings.

The dataset from Experiment 2 – consisting of *which* embedded interrogative like those in (4), repeated here as (15) – yielded virtually the same results, as shown in Figure 15. This suggests that GPT-2 small is cross-constructionally robust up four levels of clausal embedding.

(15) a. Someone wondered which lawyer(s) I think was/were ... [$N_{sg/pl}$, LEVEL1, $V_{sg/pl}$]

b. Someone wondered which lawyer(s) I think you said was/were ... [$N_{sg/pl}$, LEVEL2, $V_{sg/pl}$]

c. Someone wondered which lawyer(s) I think you said you thought was/were ...

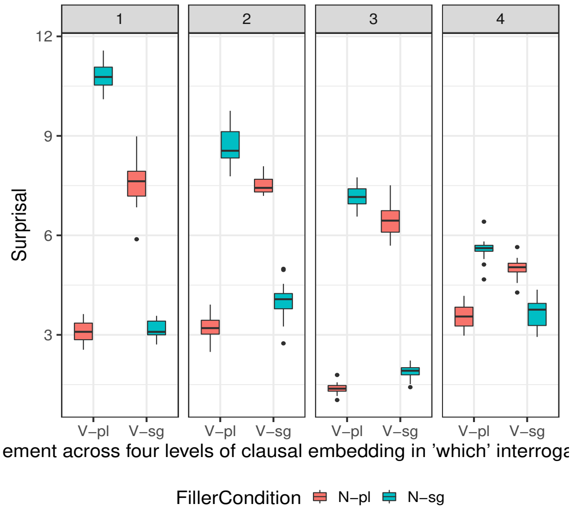


Figure 15: Surprisal of the gap-agreeing verb in 'which' questions across levels of embedding (GPT-2)

[$\mathrm{N}_{sg/pl}$, LEVEL3, $\mathrm{V}_{sg/pl}$]

d. Someone wondered which lawyer(s) who people believe I think you said you thought was/were ...
[$\mathrm{N}_{sg/pl}$, LEVEL4, $\mathrm{V}_{sg/pl}$]

For completeness, we also examined GPT-2's ability to maintain a filler-gap dependency without the interacting factor of subject-verb agreement in both clefts and interrogatives, analogously to what was done in Experiments 3 and 4. The same items were used, and as in the LSTM RNN and Transformer-XL cases we extracted the softmax activation of the word at the end of the item, after the prefix string is processed.

As Figure 16 shows, GPT-2 medium performed moderately well for the 20 cleft items (same data as Experiment 3), though the results were not as strong as BERT's. One major difference is that the surprisal of the wh-gap condition was systematically higher than that of the nwh-ngap condition. Ideally, the two should overlap. The relatively high surprisal of the wh-ngap condition is arguably due to the model maintaining expectations that the gap is further downstream in the sentence. Still, the results overall suggest that the filler-gap dependency is attended do.

The subject inversion counterpart of the 20 items (same data as Experiment 4) led to results closer to BERTs, whereby the surprisal of the wh-gap condition overlapped with that of nwh-ngap condition (all $p > 0.3$), as seen in Figure 17. In both of these experiments, the results were the same in subsequent embeddings.

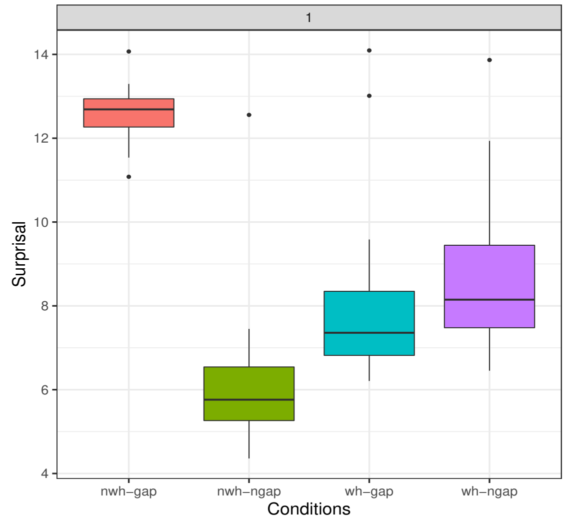


Figure 16: Surprisal of the post-gap region in uninverted indirect interrogatives in embedding 1 (GPT-2)

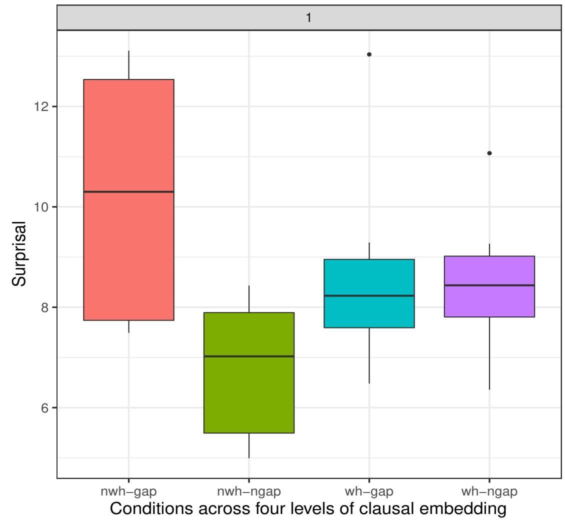


Figure 17: Surprisal of the post-gap region in inverted interrogatives across in embedding 1 (GPT-2)

7 Discussion

Filler-gap dependencies still pose challenges for general-purpose large-scale state-of-the-art neural architectures. We show LSTM RNNs fair very poorly, despite the results of Wilcox et al. (2018) and Wilcox et al. (2019b). More modern models like Transformer-XL and XLNet do even worse.

However, BERT and GPT-2 perform rather well, although not without some mixed results. For example, the performance differs significantly across different kinds of filler-gap dependency, which suggests that the models are somewhat brittle even though they are extremely large, and trained on enormous amounts of data.

References

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. ArXiv:1901.02860v3, `arxiv.org/pdf/1901.02860.pdf`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805, `http://arxiv.org/abs/1810`.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. Unpublished ms. `https://arxiv.org/pdf/1901.05287.pdf`.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL*, pages 1195–1205.

John T. Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of NAACL-2001, Pittsburg, PA*, pages 159–166. ACL.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL-HLT 2018*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Jo and Yoshua Bengio. 2017. Measuring the tendency of CNNs to learn surface statistical regularities. *CoRR*, abs/1711.11561.

Rafal Jozefowicz, Vinyals Oriol, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR*.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 3(106):1126–1177.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Tal Linzen and Brian Leonard. 2018. Distinct patterns of syntactic agreement errors in recurrent networks and humans. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 692–697. Cognitive Science Society, Austin, TX.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marten van Schijndel and Tal Linzen. 2018. Modeling garden path effects without explicit hierarchical syntax. In Tim Rogers, Marina Rau, Jerry Zhu, and Chuck Kalish, editors, *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2600–2605. Cognitive Science Society, Austin, TX.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of Blackbox NLP at ACL*, page pp.10.

Ethan Wilcox, Roger P. Levy, Takashi Morita, and Richard Futrell. 2018. What do rnn language models learn about filler-gap dependencies? In *Proceedings of the Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Ethan Wilcox, Roger P. Levy, Takashi Morita, and Richard Futrell. 2019b. What syntactic structures block dependencies in RNN language models? In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society (CogSci)*.

Thomas Wolf. 2019. Some additional experiments extending the tech report "Assessing BERT's Syntactic Abilities" by Yoav Goldberg. Unpublished ms. `huggingface.co/bert-syntax/extending-bert-syntax.pdf`.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. `arxiv.org/abs/1906.08237`.

Neural network learning of the Russian genitive of negation: optionality and structure sensitivity

Natalia Talmina
Department of Cognitive Science
Johns Hopkins University
talmina@jhu.edu

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

Abstract

A number of recent studies have investigated the ability of language models (specifically, neural network language models without syntactic supervision) to capture syntactic dependencies. In this paper, we contribute to this line of work and investigate the neural network learning of the Russian genitive of negation. The genitive case can optionally mark direct objects of negated verbs, but it is obligatory in the existential copula construction under negation. We find that the recurrent neural network language model we tested can learn this grammaticality pattern, although it is not clear whether it learns the locality constraint on the genitive objects. Our results further provide evidence that RNN models can distinguish between optionality and obligatoriness.

1 Introduction

Statistical language models are probability distributions over sequences of words, which they learn from large corpora during training. For any given context, these models assign a probability to all of its possible continuations: for a example, given the context "he was eating soup with a. . . ", language models can predict that the word "spoon" is much more likely to occur next than "shoe".

A class of language models – Recurrent Neural Network (RNN) models – have been particularly successful on various applied language tasks (Mikolov et al., 2010; Vinyals et al., 2015; Kiperwasser and Goldberg, 2016; Bahdanau et al., 2014). But what kind of linguistic knowledge do these models capture? Arguably, human language knowledge is comprised of more than word co-occurrence statistics – it encompasses abstract rules and generalizations that concern hierarchical structure. According to the argument from the poverty of the stimulus (Chomsky, 1980), the kind of structural knowledge that underlies human linguistic performance is impossible to derive purely from the input language learners receive, since many structure-dependent linguistic phenomena are too infrequent in the type of input humans encounter during language acquisition. Therefore, according to the argument, human sensitivity to the structure in language must be innate.

Since neural networks do not possess this innate bias – but perform applied natural language tasks with high accuracy – they can provide a rich source of information about the mechanisms underlying hierarchical structure rule learning. A number of questions need to be asked. How much grammar can language models learn just from a corpus? What are the limitations on the generalizations they can make about hierarchical structures? Recently, several studies have addressed these questions by testing RNNs' performance on structure-sensitive grammatical tasks. The results of these studies showed that RNNs can learn subject-verb agreement (Linzen et al., 2016; Gulordava et al., 2018; Ravfogel et al., 2018), filler-gap dependencies (Wilcox et al., 2018), hierarchical rules of question formation (McCoy et al., 2018), and the contexts that license negative polarity items (Jumelet and Hupkes, 2018).

In this paper, we contribute to this line of research by extending it to issues in Russian syntax. What makes Russian compelling is that it has rich morphology, which allows us to expand the range of tasks that have been used in previous work to explore RNN learning of structural dependencies. In particular, Russian has case-marking alternations involving the genitive case: along with the accusative case (which is typical cross-linguistically), the genitive can mark direct objects of transitive verbs. However, it is only licensed under negation, and is **optional** – the accusative case can be used in both affirmative and negative clauses. The genitive also alternates with the nominative case to mark the subjects of existential copula constructions, where it is **obligatory** under negation. Nominative subjects are only allowed with affirmative sentences. We spell out these properties in more detail in the next section.

2 Background: Russian genitive-of-negation

In Russian, direct objects are usually marked by the accusative case, as is common in languages with overt case marking:

(1) Uchitel proveril domasniye zadaniya
Teacher graded homeworksACC
"The teacher graded the homeworks."

However, non-oblique arguments can receive genitive case in the scope of sentential negation – a phenomenon known as the genitive of negation (Bailyn, 1997; Pesetsky, 1982; Paducheva, 2004; Harves, 2002; Timberlake, 1975; Babby, 1980):

(2) Uchitel **ne** proveril domasniye zadaniya
Teacher **neg** graded homeworks.ACC
"The teacher did not grade the homeworks."

(3) Uchitel **ne** proveril domasnih zadaniyj
Teacher **neg** graded homeworks.GEN
"The teacher did not grade the homeworks."

If the sentence is affirmative, only the accusative case can be used to mark the direct object:

(4) * Uchitel proveril domasnih zadaniyj
Teacher graded homeworks.GEN
"The teacher graded the homeworks."

Further, the genitive is only licensed when the negation term is local: in sentences like (5), the relative clause negation cannot license genitive case-marking on the main verb object *domasnih zadaniyj*. We will refer to this licensing pattern as the LOCALITY CONSTRAINT.

(5) * Uchitel, kotoryj ne lyubil studentov,
Teacher who **neg** like students
proveril domasnih zadaniyj
graded homeworks.GEN
"The teacher, who didn't like the students, graded the homeworks."

The genitive of negation is considered to be **optional** in sentences like (3) (Kagan 2010, although see Bailyn 1997; Harves 2002 for discussion), but it is **obligatory** in the existential copula construction, where the genitive alternates with the nominative case:

(6) (Bailyn, 1997)

a. Na stole **net** knig
on table **neg** books.GEN
"There are no books on the table."

b. * Na stole **net** knigi
on table **neg** books.NOM
"There are no books on the table."

3 Overview of experiments

Motivated by the observations in the previous section, we explored how well language models can capture the properties of the genitive of negation. We ran a series of experiments to study the behavior of an RNN language model trained by Gulordava et al. (2018). In Experiment 1, we tested the language model on simple sentences with case-marking alternation on direct objects, finding that the model learned the grammaticality pattern in (3–4). In Experiments 2–4, we tested whether the model was sensitive to the structurally defined scope of negation. We found that the model correctly predicted the genitive-accusative alternation even when there was no overt marking of sentential scope. In Experiment 5, we tested the model on the existential copula construction in which the genitive case is obligatory under negation. Our results suggest that the model could differentiate between the syntactic structures where the genitive case is obligatory from those where it is optional.

4 Methodology

To explore whether RNN language models can capture the constraints on genitive-marked direct objects, we studied the performance of the model presented in Gulordava et al. (2018). The model was trained on a 90-million-word corpus extracted from the Russian Wikipedia and had two layers of 650 hidden LSTM units. Additionally, we trained a 3-gram model on the same corpus to provide a baseline for our experiment. The 3-gram model which backs off to smaller n-grams using linear interpolation.

Following previous work (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018), we assessed the model's performance by examining the probabilities it assigned to grammatical sentences from our dataset, compared to ungrammatical ones. We used surprisal (Hale, 2001):

$$\text{surprisal}(w_i) = -\log \text{P}(w_i \mid w_1 \dots w_{i-1})$$

The higher the surprisal, the more unexpected a word is under the model's probability distribution. Since the sentences in (3) and (4) are minimally different from each other (the only difference being that the verb in (3) is negated), we can directly compare the surprisal the model assigned to the genitive-marked objects in these sentences. Assuming the probability distribution defined by the model reflects the grammar of the genitive of negation construction, we expected that the genitive-marked object would be assigned higher surprisal in (4), where it is not licensed by negation. Since accusative objects are grammatical independently of polarity, we did not expect the same difference between (1) and (2).

5 Experiments

5.1 Experiment 1: Simple sentences

5.1.1 Materials

We constructed a dataset of 64 sentences, each consisting of a subject, a verb, and an object. For each sentence, we included four versions which varied in main verb polarity (positive or negative) and the case marking of the direct object (accusative or genitive), yielding a total of 256 experimental items. Examples (7a–7d) represent all four conditions for one item in our dataset. Only the sentence in (7b) is ungrammatical: both (7a) and (7c) are grammatical because accusative objects are always licensed, and in (7d), the genitive of negation is grammatical because it is within the scope of a negated verb. In (7b), however, the genitive-marked object is not licensed by negation, which makes the whole sentence ungrammatical.

(7) a. **positive-accusative**

Vystavka artista poterpela proval
Exhibition of-artist suffered failure.ACC

"The artist's exhibition was a failure."

b. **positive-genitive**

* Vystavka artista poterpela
Exhibition of-artist suffered
provala
failure.GEN

"The artist's exhibition was a failure."

c. **negative-accusative**

Vystavka artista **ne** poterpela
Exhibition of-artist **neg** suffered
proval
failure.ACC

"The artist's exhibition wasn't a failure."

d. **negative-genitive**

Vystavka artista **ne** poterpela
Exhibition of-artist **neg** suffered
provala
failure.GEN

"The artist's exhibition wasn't a failure."

Given this pattern, we expected that the model would assign higher surprisal to the word *provala* 'failure.GEN' in (7b) than in (7d), but there would be no such difference for the word *proval* 'failure.ACC' in (7a) and (7c).

5.1.2 Results

LSTM Consistent with our predictions, the genitive-marked direct objects were less surprising when the verb was negated (see Figure 2a). Figure 3a shows that the difference between the positive and negative conditions is much bigger for genitive-marked objects than for the accusative-marked ones. This suggests the model learned that the negative-polarity constraint only applies to objects marked by the genitive case.

We further tested this by running a linear mixed effects model (Baayen et al., 2008) with the model-assigned surprisal as the dependent variable, and case, polarity, their interaction, and item frequency as predictors. We found a main effect of case ($p = 0.004$), as well as an interaction between case and polarity ($p < 0.0001$). Surprisal was significantly affected by polarity for genitive-marked objects ($p < 0.0001$), but not for accusative objects ($p = 0.09$).

Although we did not find a main effect of frequency, we performed a follow-up analysis aimed to rule out the possibility that unigram frequency could be a confound for these results. Figure 1 shows that accusative-marked objects in our dataset had much higher unigram frequency in the training corpus than the genitive-marked objects. To test for the presence of the frequency effects, we re-ran the linear mixed effects analysis on surprisal scores that we normalized by subtracting the target word's log frequency from its surprisal score. The pattern remained the same: we found main effects of frequency ($p = 0.006$) and, as before, of case ($p = 0.004$), as well as an interaction between case and polarity ($p < 0.0001$).

N-gram We found a main effect of case ($p < 0.0001$) and frequency ($p = 0.001$), but not of polarity ($p = 0.7$). There was no interaction between case and polarity ($p = 0.8$). Figure 4b shows there was no difference between the positive and negative conditions for either case. We observed this pattern in all experiments we ran, unless otherwise stated.

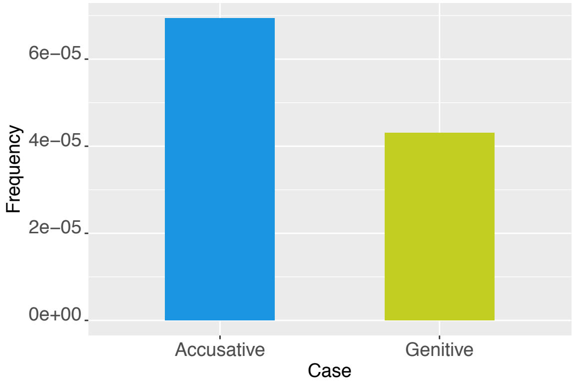


Figure 1. Average unigram frequency (word count divided by the size of the training corpus) of accusative and genitive objects from our dataset.

5.1.3 Discussion

Our results suggest that the model at least learned to encode case: to predict the grammaticality pattern in (7a–7d), the model needed to infer that the grammaticality of the genitive case – but not the accusative – is constrained by the presence of negation.

However, these results alone are not sufficient to conclude that the model was able to infer the syntactic structure that licenses the genitive of negation. Since our experimental items had SVO word order, it could have instead learned a linear rule where the genitive-marked object is allowed whenever it follows negation. Instead, the locality constraint would predict that the object in the genitive case is licensed only when it is in the scope of negation.

To test whether the model has learned the locality constraint, we ran a series of experiments in which we modified our experimental sentences to include the following distractors: (1) a negated relative clause, while the genitive-marked object was licensed by the negated main clause verb, (2) a complement clause, whose polarity varied between positive and negative, and whose main clause was always negative, and (3) a negated participial construction. We give a detailed description of these constructions in the following sections.

5.2 Experiment 2: Relative clauses

5.2.1 Materials

To test whether the model learned that the genitive of negation is only licensed under the scope of sentential negation, we modified the simple sentences from our dataset to include a relative clause with a negated verb. It is crucial for the model to infer the syntactic structure of these sentences: the model needs to be able to represent local scope in order to correctly predict that (8b) is ungrammatical – since the genitive-marked object in this case is outside the scope of negation.

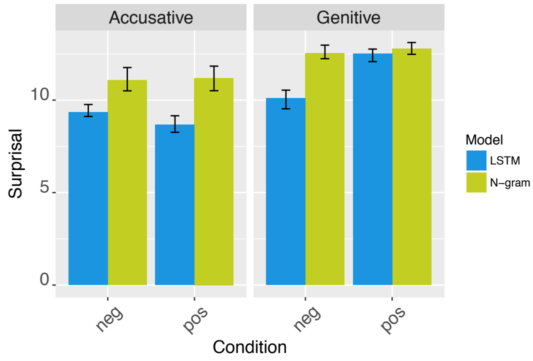
(8) a. * Vystavka artista, kotoryj **ne** lyubil
Exhibition of-artist who **neg** loved
vnimaniya publiki, poterpela provala
attention public suffered failure.GEN
"The exhibition of the artist, who didn't like public attention, was a failure."

b. Vystavka artista, kotoryj **ne** lyubil
Exhibition of-artist who **neg** loved
vnimaniya publiki, **ne** poterpela
attention public **neg** suffered
proval
failure.GEN
"The exhibition of the artist, who didn't like public attention, was not a failure."

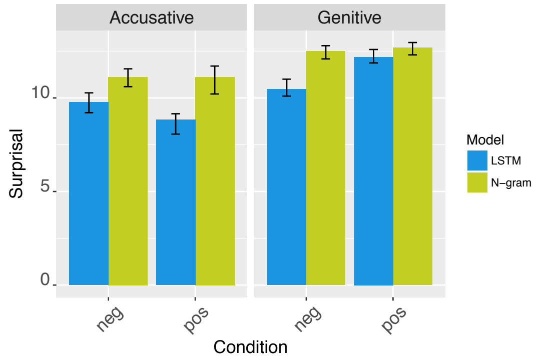
5.2.2 Results

LSTM The model's surprisal was highest in the positive-genitive condition (Figure 2b), suggesting that genitive-marked direct objects were more expected when they were licensed by the negated main clause verb. We found main effects of case ($p = 0.01$) and polarity ($p = 0.04$), and the two terms interacted ($p < 0.0001$). Polarity significantly affected both genitive-marked ($p = 0.0001$) and accusative-marked ($p = 0.04$) objects. Figure 3b shows that for the accusative-marked objects, the difference between positive and negative conditions was the inverse of the genitive case: an accusative-marked object was more surprising when the main clause verb was negated.

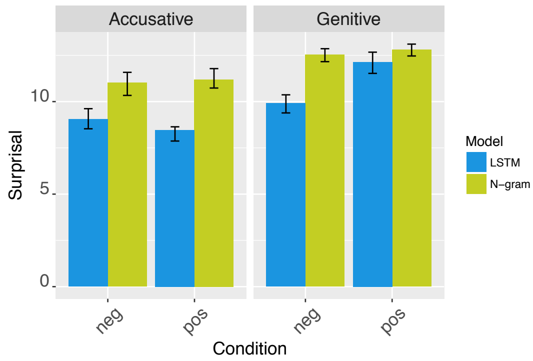
The analysis of frequency effects revealed that normalized surprisal scores were significantly af-



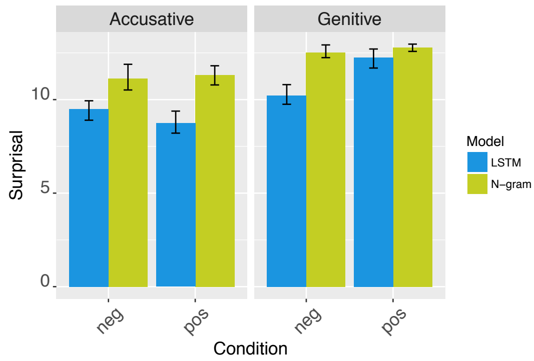
(a) Simple sentences



(b) Relative clause



(c) Complement clause



(d) Participial construction

Figure 2. Surprisal averaged by condition (Experiments 1–4). Error bars indicate 95% confidence intervals.

fected by case ($p = 0.01$), frequency ($p = 0.001$), and the interaction of case and polarity ($p < 0.0001$).

N-gram The trigram model's performance was the same as in Experiment 1.

5.2.3 Discussion

Our results suggest the model learned the genitive-marked object was licensed only when it appeared in the scope of negation – which in turn required the representation of syntactic structure. If the model had learned only the linear rule, it would have assigned the same surprisal in both positive-genitive and negative-genitive conditions, since both linearly followed the negation in the scope of the relative clause.

The main effect of polarity suggests that the model possibly learned an interaction between case and polarity, preferring accusative objects with affirmative sentences and genitive objects under negation.

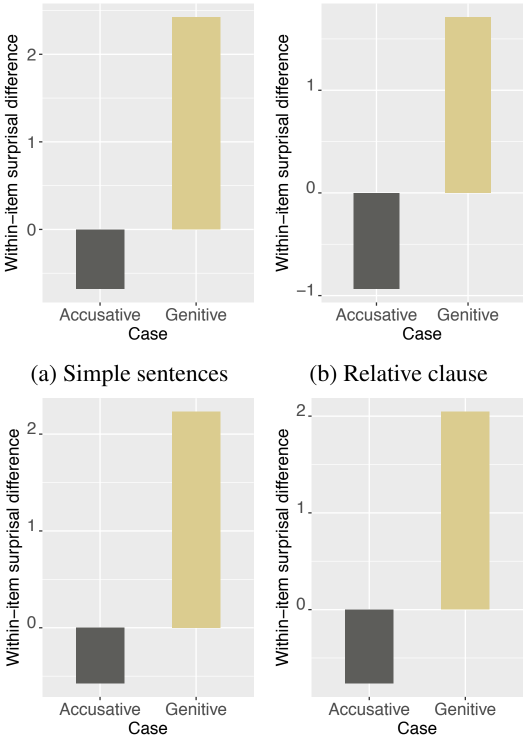
5.3 Experiment 3: Complement clauses

5.3.1 Materials

In the previous experiment, the distractor (i.e. the negation term that needed to be ignored) was always in the relative clause. This implies that there are two possible interpretations of the results: 1) the model could represent the scope of negation and apply it to the genitive licensing rule, or 2) the model learned to ignore negation if it immediately followed the word *kotoryj* 'that/who', which marked the beginning of an embedded clause. To rule out the second possibility, we tested the model's performance on sentences with complement clauses. In this set of sentences, the distractor was in the main clause, while the target word (the accusative- or genitive-marked direct object) was in an embedded clause. The embedded clause varied between positive and negative polarity – and only the latter licensed the genitive object:

(9) a. * Zhurnalist **ne** znal chto vystavka
Journalist **neg** knew that exhibition
artista poterpela provala
of-artist suffered failure.GEN
"The journalist didn't know that the artist's exhibition was a failure."

b. Zhurnalist **ne** znal chto vystavka
Journalist **neg** knew that exhibition
artista **ne** poterpela provala
of-artist **neg** suffered failure.GEN



(a) Simple sentences (b) Relative clause

(c) Complement clause (d) Participial construction

Figure 3. Within-item difference between positive and negative conditions, averaged by case (Experiments 1–4).

"The journalist didn't know that the artist's exhibition was not a failure."

5.3.2 Results

LSTM Average surprisal was lower for genitive-marked objects when the embedded clause contained a negated verb (Figure 2c), suggesting the model learned to represent sentential scope and did not mistake main clause negation for a licensor. Average within-item difference between positive and negative conditions was also greater for the genitive case (Figure 3c).

As before, we ran a linear mixed effects model to test the significance of these findings. We found a main effect of case ($p = 0.0006$), as well as an interaction between case and polarity ($p < 0.0001$). The surprisal the language model assigned to genitive-marked objects was significantly affected by the embedded clause's polarity ($p < 0.0001$), while there was no such effect for the accusative case ($p = 0.17$).

Our analyses of surprisal scores normalized by frequency revealed main effects of case ($p = 0.0004$) and frequency (0.002), as well as an interaction between case and polarity ($p < 0.0001$).

N-gram The model's performance was the same as in Experiment 1.

5.3.3 Discussion

These results provide further evidence that the model learned the locality constraint on genitive licensing: although the main clause verb was negated in all four conditions, the surprisal the model assigned to the genitive-marked object was reduced when the verb in the embedded clause was negated as well.

5.4 Experiment 4: Participial constructions

5.4.1 Materials

Experiments 2 and 3 provide some evidence that the model learned the scope constraint on the genitive of negation. However, the sentences we tested in these experiments contained overt cues that indicated the scope of negation that the model needed to ignore: in Experiment 1, the relative pronoun *kotoryj* indicates the beginning of the relative clause, and in Experiment 2, the pronoun *chto* indicates the beginning of the complement clause. Would the model be able to identify the scope of negation without these cues? We investigated this by testing the model's performance on the Russian participial construction, which has no overt function words marking the scope of negation. We constructed an experimental set of sentences which consisted of simple sentences such as those in (7a-7d) with an active present or past participle modifying the subject.

(10) a. * **Ne** poluchivshaya vnimaniya pressy
neg received.PTCP attention of-press
vystavka artista poterpela provala
exhibition of-artist suffered failure.GEN
"The artist's exhibition, which did not receive attention from press, was a failure."

b. **Ne** poluchivshaya vnimaniya pressy
neg received.PTCP attention of-press
vystavka artista **ne** poterpela
exhibition of-artist **neg** suffered
provala
failure.GEN
"The artist's exhibition, which did not receive attention from press, was not a failure."

In (10a), the genitive-marked object *provala* 'failure' is outside of the scope of negation, so we expected that it would be more surprising than in

(10b), where the genitive is licensed by sentential scope.

5.4.2 Results

LSTM Figure (2d) shows the model assigned higher probability to genitive-marked objects when they were licensed by a negated verb. A linear mixed effects analysis confirmed surprisal was affected by case ($p = 0.01$), as well as the interaction between case and polarity ($p < 0.0001$). Polarity was significant for genitive-marked objects ($p < 0.0001$), but not for accusative-marked ones ($p = 0.098$).

Surprisal scores normalized by frequency were significantly affected by case ($p = 0.01$), frequency ($p = 0.003$), and the interaction between polarity and case ($p < 0.0001$).

N-gram The model's performance was the same as in Experiment 1.

5.4.3 Discussion

The model was able to capture the grammaticality pattern in (10a–10b) despite the lack of overt scope marking cues – suggesting that the model in fact represents the scope of negation instead of relying on cues such as function words introducing embedded clauses.

5.5 Experiment 5: Existential copula construction

5.5.1 Materials

In the experiments we have presented so far, the genitive case was always optional: genitive-marked direct objects were only grammatical in the scope of sentential negation, while the accusative case was licensed whether the sentence had positive or negative polarity. We expected to see higher surprisal for genitive-marked objects when they were outside of the scope of negation, but we did not expect any polarity-related difference for the accusative case.

The situation is different in the Russian existential copula construction. First, in this construction the case alternation concerns the subject, which can be assigned the nominative or the genitive case. Second, the genitive case is always obligatory under negation. Finally, the nominative case marking is also constrained (unlike the accusative with direct objects): subjects can only receive nominative case when the sentence is affirmative. In other words, although in previous examples only the positive genitive condition was ungrammatical, in the case of the existential construction the negative nominative condition is ungrammatical as well:

(11) a. U vystavki byl proval
At exhibition was failure.NOM
"The exhibition was a failure."

b. * U vystavki byl provala
At exhibition was failure.GEN
"The exhibition was a failure."

c. * U vystavki **ne** bylo proval
At exhibition **neg** was failure.NOM
"The exhibition was not a failure."

d. U vystavki **ne** bylo provala
At exhibition **neg** was failure.GEN
"The exhibition was not a failure."

5.5.2 Results

LSTM A linear mixed-effects analysis revealed main effects of polarity ($p < 0.0001$), case ($p < 0.0001$), and frequency ($p = 0.0003$). The interaction between case and polarity was significant as well ($p < 0.0001$).

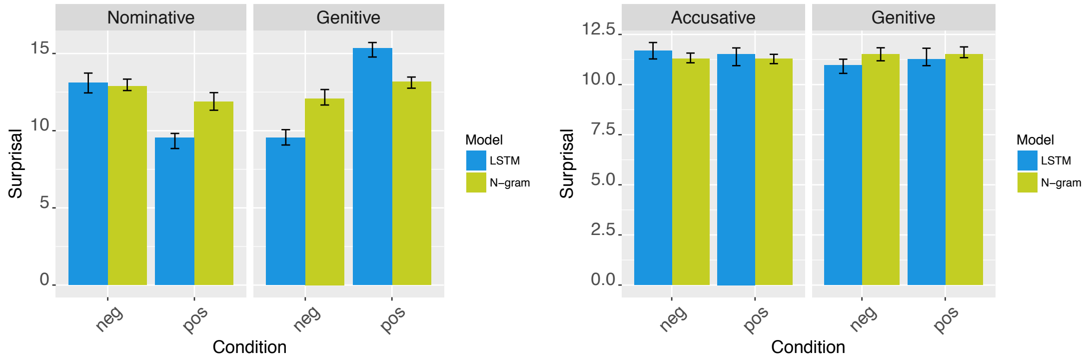
N-gram We found main effects of polarity ($p = 0.001$), case ($p = 0.0007$), and frequency ($p < 0.0001$). There was also a significant interaction of case and polarity ($p < 0.0001$).

5.5.3 Discussion

The main effect of polarity shows that the model learned constraints on both the nominative and the genitive case: the genitive is licensed under negation and ungrammatical in affirmative sentences, while the opposite is true for the nominative.

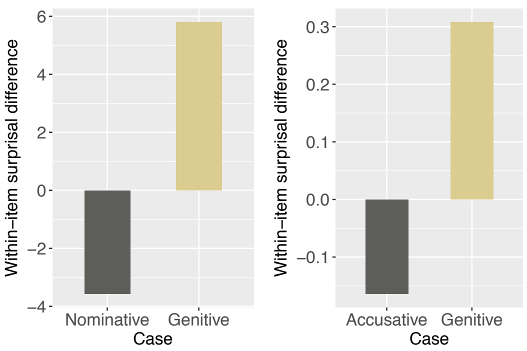
Further, within-item difference for both the nominative and the genitive is much bigger than in other experiments (Figure 5a) – which suggests that the model distinguished between optionality and obligatoriness. I.e., the magnitude of surprisal was reduced in the positive-genitive condition when it was optional under negation. However, when it was required under negation, genitive-marking with positive polarity was more surprising.

Compared to previous experiments, there was a stark difference in surprisal scores between positive and negative conditions. This could be due to the fact that the the verb *byt'* 'to be' always appears in 3rd person singular under negation, which could have provided the model with an additional cue that the genitive case is required.



(a) Existential construction (b) Parenthetical

Figure 4. Surprisal averaged by condition (Experiments 5–6). Error bars indicate 95% confidence intervals.



(a) Existential construction (b) Parenthetical

Figure 5. Within-item difference between positive and negative conditions, averaged by case (Experiments 5–6).

5.6 Experiment 6

5.6.1 Materials

In the grammatical sentences used in Experiments 1–5, the genitive objects were directly preceded by the **neg** + main verb bigram, which left open the possibility that the LSTM model relied on this linear structure as a cue that the genitive case was licensed. We constructed a new dataset where the main verb was separated from the direct object by a parenthetical (e.g. "to the surprise of the press" in 12a-12b). If the model is learning the locality rule correctly, this parenthetical should not intervene with inferring the grammaticality pattern in 12a-12b.

(12) a. * Vystavka artista poterpela, k
Exhibition of-artist suffered to
udivleniju pressy, provala
surprise of-press failure.GEN
"The artist's exhibition was a failure, to the surprise of the press."

b. Vystavka artista **ne** poterpela, k
Exhibition of-artist **neg** suffered to
udivleniju pressy, provala
surprise of-press failure.GEN
"The artist's exhibition wasn't a failure, to the surprise of the press."

5.6.2 Results

LSTM We found a main effect of case ($p < 0.0004$) and frequency ($p = 0.01$), but not of polarity ($p = 0.6$); there was no interaction between case and polarity ($p = 0.1$). Figure 4b shows there was almost no difference in surprisal the model assigned to the genitive objects licensed by negation compared to those that were ungrammatical.

N-gram There was a main effect of frequency ($p < 0.0001$), but not of case ($p = 0.34$) or polarity ($p = 0.96$). There was no interaction between case and polarity (0.97).

5.6.3 Discussion

In (12b), the negation term was local to the target genitive object, but linearly separated from it. If the model was correctly learning the locality constraint, it would be able to predict that the genitive object *provala* is grammatical in (12a), but not (12b). However, the model could not identify the negation term as the licensor in these types of sentences, assigning similar surprisal to the genitive objects in (12a) and (12b). This result, however, may be due to the rarity of the parenthetical sentences in the training corpus, and does not necessarily imply the model was not learning the constraint in Experiments 1–5.

6 General discussion and future work

In this paper, we have examined the ability of an RNN language model to learn several properties of the Russian genitive of negation. The genitive of negation can **optionally** mark direct objects of transitive verbs when the latter are negated, and is **obligatory** with subjects of existential copula constructions under negation.

To be able to learn the polarity constraint on the genitive case, the model needed to represent the scope of negation. In Experiments 2 and 3, we tested this by introducing distractors to our experimental items: negated relative clauses and complement clauses that were not licensed by sentential negation. We found that the model's performance matched our predictions, assigning higher surprisal to those genitive-marked objects that were outside of the scope of negation. The results from Experiment 4 further suggest that the model could represent the scope of negation without relying on such cues as function words explicitly marking clause boundaries.

Our results from Experiment 5 provide some evidence that the model could differentiate between optionality and obligatoriness. First, we found that both the nominative and the genitive case were significantly impacted by polarity (while only the genitive was affected in other types of sentences we tested). Second, for both the nominative and the genitive case the average within-item difference between positive and negative conditions was much bigger than in other experiments. Taken together, these results suggest that the model learned that the genitive of negation was obligatory in existential sentences.

The results of Experiment 6 reveal that the model could not learn the locality constraint on the genitive of negation when the linear distance between the main verb and the direct object was increased. We tested sentences where a parenthetical intervened before the main verb and its object, and the model did not differentiate between the sentences in which the genitive object was licensed by a local negation term from those where it was not. However, this finding does not necessarily imply that the model did not learn the locality constraint in Experiments 1–5. One possible explanation for the model's behavior on the task in Experiment 6 is that constructions where a parenthetical intervenes between the main verbs and its object are not frequent in a natural corpus.

Further, more evidence is needed to asses whether the model could differentiate between syntactic structures which optionally licensed the genitive case from those where it was obligatory. One limitation of our approach is that we used the same metric for both optional and obligatory uses of the genitive of negation: we compared the surprisal the model assigned to grammatical and ungrammatical sentences, and the negated sentences with the genitive case were grammatical whether the genitive was obligatory or optional. A possible direction for future work could involve a comparison of our results to human processing data (e.g. as in Futrell and Levy 2018). Since surprisal scores tend to correlate with reaction times (Smith and Levy, 2013), we would expect our results to match human performance.

Finally, our study only addressed some properties of the genitive of negation and only a subset of the syntactic structures in which it can appear. We haven't looked, for instance, into the genitive case marking of unaccusative subjects (13) and derived subjects of passives (14) (Bailyn, 1997):

(13) (Babby, 1980)

Zdes' ne rastet gribov
here **neg** grows mushrooms.GEN

"No mushrooms grow here."

(14) (Bailyn, 1997)

Ne bylo polucheno gazet
neg was received newspapers.GEN

"No newspapers were received."

There is also a slight difference in meaning between the genitive and accusative direct objects that we haven't addressed: while accusative direct objects usually receive a definite interpretation, the genitive ones have an existential or indefinite interpretation (Bailyn, 1997; Harves, 2002).

While future investigation into these issues is needed to gain a full picture of neural network learning of the genitive of negation, our study adds to the growing body of evidence that RNN language models do not need syntactic supervision or a hierarchical bias to capture syntactic dependencies. Whether the same is true for human language learners remains to be seen.

References

R Harald Baayen, Douglas J Davidson, and Douglas M Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language*, 59(4):390–412.

Leonard H Babby. 1980. *Existential Sentences and Negation in Russian*. Karoma Publishers: Ann Arbor, MI.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *International Conference for Learning Representations*.

John F Bailyn. 1997. Genitive of negation is obligatory. In *Formal approaches to slavic linguistics*, volume 4, pages 84–114. University of Michigan Press: Ann Arbor, MI.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2014. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*.

Noam Chomsky. 1980. Rules and representations. *Behavioral and brain sciences*, 3(1):1–15.

Richard Futrell and Roger P Levy. 2018. Do RNNs learn human-like abstract word order preferences? *arXiv preprint arXiv:1811.01866*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Stephanie Harves. 2002. Genitive of negation and the syntax of scope. In *Proceedings of ConSOLE*, volume 9, pages 96–110.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? On the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Olga Kagan. 2010. Genitive objects, existence and individuation. *Russian linguistics*, 34(1):17–39.

Nina Kazanina. 2017. Predicting complex syntactic structure in real time: Processing of negative sentences in Russian. *The Quarterly Journal of Experimental Psychology*, 70(11):2200–2218.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018*, pages 1192–1202.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2093—2098, Austin, TX.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Elena V Paducheva. 2004. The genitive subject of the verb byt' (to be). *Studies in Polish linguistics*, 1:47–59.

David Pesetsky. 1982. *Paths and categories*. Ph.D. thesis, MIT.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM learn to capture agreement? The case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Alan Timberlake. 1975. Hierarchies in the genitive of negation. *The Slavic and East European Journal*, 19(2):123–138.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler-gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Modeling Morphological Processing in Human Magnetoencephalography

Yohei Oseki
Faculty of Science & Engineering
Waseda University
oseki@aoni.waseda.jp

Alec Marantz
Department of Linguistics & Psychology
New York University
marantz@nyu.edu

Abstract

In this paper, we conduct a magnetoencephalography (MEG) lexical decision experiment and computationally model morphological processing in the human brain, especially the Visual Word Form Area (VWFA) in the visual ventral stream. Five neurocomputational models of morphological processing are constructed and evaluated against human neural activities: Character Markov Model and Syllable Markov Model as "amorphous" models without morpheme units, and Morpheme Markov Model, Hidden Markov Model (HMM), and Probabilistic Context-Free Grammar (PCFG) as "morphous" models with morpheme units structured linearly or hierarchically. Our MEG experiment and computational modeling demonstrate that "morphous" models outperformed "amorphous" models, PCFG was most neurologically accurate among "morphous" models, and PCFG better explained nested words with non-local dependencies between prefixes and suffixes. These results strongly suggest that morphemes are represented in the human brain and parsed into hierarchical morphological structures.

1 Introduction

Under the single-route decomposition model of morphologically complex visual word recognition (Taft, 1979, 2004; Taft and Forster, 1975), there are three functionally different stages of morphological processing: morphological decomposition, lexical access, and morphological recombination. In the first stage of morphological decomposition, morphologically complex words are visually decomposed into component morphemes. In the second stage of lexical access, meanings of decomposed morphemes are lexically retrieved from the mental lexicon. In the last stage of morphological recombination, retrieved meanings of decomposed morphemes are semantically composed.

In the cognitive neuroscience literature, Fruchter and Marantz (2015) employed magnetoencephalography (MEG) to spatiotemporally dissociate those stages of morphological processing. Specifically, the first stage of morphological decomposition has been indexed by evoked response components such as M170 (Zweig and Pylkkänen, 2009; Solomyak and Marantz, 2010; Lewis et al., 2011; Fruchter et al., 2013; Gwilliams et al., 2016) or Type II (Tarkiainen et al., 1999; Helenius et al., 1999) in the visual ventral stream of the human brain (Pylkkänen and Marantz, 2003; Hickok and Poeppel, 2007). Moreover, Dehaene et al. (2005) proposed local combination detectors (LCDs) where linguistic units such as characters, syllables, and morphemes are convolutionally represented and processed in the visual ventral stream from posterior occipital to anterior temporal cortices and, importantly, morphemes have been localized to the left fusiform gyrus known as the Visual Word Form Area (VWFA; Cohen et al., 2000, 2002; Dehaene et al., 2001, 2002). For example, Solomyak and Marantz (2010) and Lewis et al. (2011) computed transition probabilities from stems to suffixes (e.g. $P(\text{Suffix}|\text{Stem})$) to successfully predict neural responses to real (e.g. *teach-er*) and pseudo (e.g. *corn-er*) bimorphemic words, respectively. These results have suggested that morphemes may be neurologically real in the human brain.

However, "amorphous" models without morpheme units have recently been proposed in the morphological processing literature (Baayen et al., 2011; Virpioja et al., 2017). For instance, Baayen et al. (2011) and Milin et al. (2017) proposed Naive Discriminative Learning (NDL), a connectionist model with direct mappings from forms to meanings, to explain morphological processing without morpheme units. In addition, Virpioja et al. (2017) and Hakala et al. (2018) employed

Morfessor, an unsupervised finite-state model with statistically induced "morphs" (Creutz and Lagus, 2007), to predict human reaction times and neural responses without linguistically defined morphemes. Furthermore, as correctly pointed out by Libben (2003, 2006), bimorphemic words exclusively tested in the previous literature (Zweig and Pylkkänen, 2009; Solomyak and Marantz, 2010; Lewis et al., 2011) cannot distinguish linear morphological decomposition from hierarchical morphological parsing (cf. Song et al., 2019; Oseki et al., 2019). Therefore, whether morphemes are represented in the human brain and, if so, processed linearly or hierarchically remains to be empirically investigated.

In this paper, we conduct an magnetoencephalography (MEG) experiment where participants perform visual lexical decision on morphologically complex words and, generalizing the computational modeling technique developed in the sentence processing literature (Frank et al., 2015; Brennan et al., 2016), computationally model morphological processing in the human brain, with special focus on the VWFA in the visual ventral stream. Specifically, five neurocomputational models of morphological processing are constructed and evaluated against human neural activities: Character Markov Model and Syllable Markov Model as "amorphous" models without morpheme units, and Morpheme Markov Model, Hidden Markov Model (HMM), and Probabilistic Context-Free Grammar (PCFG) as "morphous" models with morpheme units structured linearly or hierarchically.

2 Methods

2.1 Participants

The participants were 26 native English speakers recruited at New York University. All participants were right-handed according to the Edinburgh Handedness Inventory (Oldfield, 1971) and with normal or corrected-to-normal vision. They provided written informed consent and were paid $15/hour for their participation. We excluded 6 participants based on their behavioral performance: 3 participants excluded due to low accuracy ($< 75\%$) and 3 participants excluded due to slow (> 2000 ms) or fast mean reaction times ($<$ 500 ms). Thus, 20 participants were included in the statistical analyses (10 males and 10 females, $M = 28.4$, $SD = 9.27$).

2.2 Stimuli

The stimuli were 800 morphologically complex trimorphemic words and nonwords. The stimuli creation procedure consisted of several steps. First, 600 trimorphemic words were created based on the CELEX database (Baayen et al., 1995) in accordance with syntactic (syntactic categories), morphological (affix combinations), and phonological (orthographic adjustments) selectional restrictions of derivational affixes, but without semantic selectional restrictions explicitly taken into consideration. In this sense, these trimorphemic words are grammatical ("possible") but not necessarily acceptable ("actual") words (cf. Halle, 1973; Bauer, 2014). These 600 trimorphemic words were subcategorized into 300 linear words $[_X\ [_Y\ [_Z\ \sqrt{\text{Root}}]\ \text{Suffix}]\ \text{Suffix}]$ with productive derivational suffixes (Plag and Baayen, 2009) and 300 nested words $[_X\ \text{Prefix}\ [_Y\ [_Z\ \sqrt{\text{Root}}]\ \text{Suffix}]]$ with productive derivational prefixes (Zirkel, 2010). Furthermore, these trimorphemic words have zero surface frequencies in the CELEX database, thereby enhancing the possibility that those words have never been encountered by participants and stored in the mental lexicon (Hay, 2003). Second, in order to weed out semantically implausible words, 600 trimorphemic words were normed with crowdsourced acceptability judgment experiments, where participants judged them on 1$\sim$7 Likert scale. Third, 500 trimorphemic words (250 linear and 250 nested) with higher acceptability judgments (> 3.5) and lower standard deviations (< 2.5) were selected and, correspondingly, 500 trimorphemic nonwords (250 linear and 250 nested) were also created based on the CELEX database in violation of syntactic selectional restrictions of inner derivational suffixes, resulting in 1000 trimorphemic words and nonwords. Fourth, in order to ensure that words and nonwords are correctly judged as such, 1000 trimorphemic stimuli were further normed with crowdsourced lexical decision experiments, where participants decided whether presented stimuli were possible English words or not as quickly and accurately as possible. Finally, 400 trimorphemic words (200 linear and 200 nested) and 400 trimorphemic nonwords (200 linear and 200 nested) with higher accuracies ($> 75\%$) were selected, resulting in the balanced and extensively normed set of 800 trimorphemic stimuli to be tested in this experiment. The stimuli are summarized in Table 1:

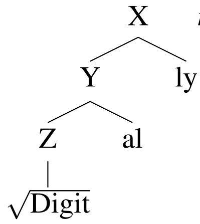
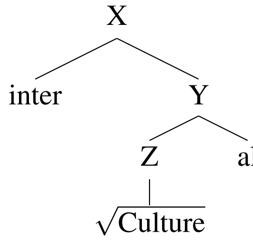
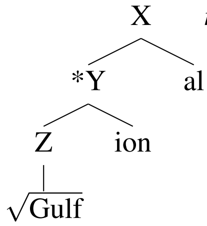
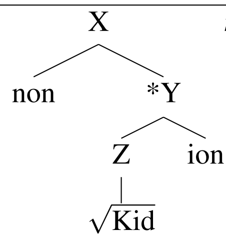
	Linear	Nested
Word	 $n = 200$	 $n = 200$
Nonword	 $n = 200$	 $n = 200$

Table 1: Summary of stimuli. The horizontal dimension is morphological structure: linear vs. nested. The vertical dimension is lexicality status: word vs. nonword. The asterisk (*) on subtrees (Y) of nonwords indicates that inner derivational suffixes violate syntactic selectional restrictions on syntactic categories of roots.

2.3 Procedure

The experiment was conducted in the Neuroscience of Language Lab at New York University, New York. Before MEG recording, each participant's head shape was digitized with a Polhemus FastSCAN laser scanner (Polhemus, Vermont, USA) and five fiducial points were marked on his/her forehead, onto which marker coils were attached during the recording. In order to familiarize the participants with visual lexical decision, the participants completed one practice block with 16 practice stimuli, 4 stimuli per each stimulus type, that do not overlap with the target stimuli. The task instructions were exactly the same as the main experiment, but the participants received feedback after each trial ("CORRECT" or "INCORRECT") during the practice block.

A 157-channel axial gradiometer whole-head MEG system (Kanazawa Institute of Technology, Kanazawa, Japan) recorded the MEG data continuously at a sampling rate of 1000 Hz (1 datapoint per each millisecond), while the participants lay in a dimly lit magnetically shielded room (MSR) and performed visual lexical decision. The MEG data were filtered online between DC and 200 Hz with a notch filter at 60 Hz. Five marker coils were attached to the corresponding fiducial points marked on the forehead and their positions were measured before and after the main experiment, in order to align the MEG data and head shapes and estimate how much the participants moved during the MEG recording. The main experiment itself lasted for about 35 minutes.

The stimuli were presented with PsychoPy package (Peirce, 2007, 2009) in Python. They were projected on the screen approximately 50 cm away from the participants and presented in white 30 lowercase Courier New font on a grey background. The 800 stimuli were randomly distributed into 8 blocks of 100 stimuli with 25 stimuli from each stimulus type. First, the explanation appeared on the screen: "In this experiment, you will read English words and determine whether you think they are possible English words. We are not concerned with whether or not these words are actual English words already listed in a dictionary. Instead, we are interested in whether or not these words could be used by a native speaker of English". Then, the task instruction appeared on the screen: "The experiment is about to begin. Please fixate on the cross in the center of the screen. Respond with your index finger if the string is word. Respond with your middle finger if it is not a word". Each trial consisted of the fixation cross (+) for 500 ms, the blank for 300 ms, and the stimulus until the participants respond with their index finger (YES) or middle finger (NO) of their left hand. The inter-stimulus interval (ISI) followed the standard normal distribution with the mean of 400 ms and the standard deviation of 100 ms.

2.4 Computational models

Five computational models were implemented with Natural Language Tool Kit package (Bird et al., 2009) in Python: Character Markov Models (Character), Syllable Markov Models (Syllables), Morpheme Markov Models (Markov), Hidden Markov Model (HMM), and Probabilistic Context-Free Grammar (PCFG). Those models were trained on the entire CELEX database via Maximum Likelihood Estimation with token weighting and Lidstone smoothing at $\alpha = 0.1$. The architectures of Markov Model, HMM, and PCFG are summarized below.

2.4.1 Markov Model

Markov Models (also called *n*-gram models) are defined by *n*-order Markov processes that compute transition probabilities of linguistic units (e.g. characters, syllables, morphemes) at position *i* given *i*–*n* context (e.g. $P(x_i|x_{i-n}, x_{i-1})$). Since the length of morphologically complex words is inherently limited relative to syntactically complex sentences, Markov Models were defined with n = 1 (i.e. bigram models), which compute transition probabilities of linguistic units at position *i* given the immediately preceding unit (e.g. $P(x_i|x_{i-1})$). For training, Markov Models were trained on character strings (Character Markov Model), syllable strings (Syllable Markov Model), and morpheme strings (Morpheme Markov Model), respectively, where character and morpheme strings were available from the CELEX database, while syllable strings were generated with `syllabify` module implemented in Python by Kyle Gorman through ARPABET transcriptions assigned by LOGIOS Lexicon Tool in the Carnegie Mellon University Pronouncing Dictionary. For testing, those trained Markov Models then computed morpheme probabilities of morphologically complex words equivalent to their transition probabilities given the Markov assumption. Markov Models are linear models, which should accurately predict local dependencies of linear words (e.g. *digitally*), but not non-local dependencies of nested words (e.g. *unpredictable*) because local dependencies (e.g. **unpredict*) are unattested in the training data.

2.4.2 Hidden Markov Model

HMMs generalize Markov Models with *n*-order Markov processes defined over "hidden" linear strings. HMMs compute transition probabilities of part-of-speech (POS) tags at position *i* given *i*–*n* context (e.g. $P(t_i|t_{i-n}, t_{i-1})$), and emission probabilities of morphemes at position *i* given POS tags at the same position *i* (e.g. $P(m_i|t_i)$). Like Markov Models, HMMs were also defined with *n* = 1, which compute transition probabilities of POS tags at position *i* given the immediately preceding POS tag (e.g. $P(t_i|t_{i-1})$). For training, HMMs were supervisedly trained on tagged morpheme strings generated from morphological structures available from the CELEX database (e.g. [(*digit*, N), (*al*, A), (*ly*, B)]). For testing, those trained HMMs then computed morpheme probabilities of morphologically complex words as the ratio of prefix probabilities at position *k* to position *k*–1, where prefix probabilities are the sum of path probabilities compatible with morphemes until position *k* (Rabinar, 1989). HMMs are linear models, which should accurately predict local dependencies of linear words (e.g. N-A-B for *digitally*), but also non-local dependencies of nested words (e.g. *unpredictable*) if component local dependencies (e.g. A-V for **unpredict*) are attested in the training data.

2.4.3 Probabilistic Context-Free Grammar

PCFGs generalize Context-Free Grammars (CFGs) with probability distributions defined over hierarchical structures. PCFGs compute nonterminal probabilities of right-hand sides given left-hand sides of nonterminal production rules (e.g. $P(rhs|lhs)$), and terminal probabilities of right-hand side terminals given left-hand side nonterminals of terminal production rules (e.g. $P(m_i|t_i)$), equivalent to HMM emission probabilities. Nonterminal production rules are head-lexicalized, which model syntactic selectional restrictions of derivational affixes (e.g. N $\rightarrow$ A *ness*). For training, PCFGs were supervisedly trained on morphological structures available from the CELEX database (e.g. [$_{\text{B}}$ [$_{\text{A}}$ [$_{\text{N}}$ *digit*] *al*] *ly*]). For testing, those trained PCFGs then computed morpheme probabilities of morphologically complex words as the ratio of prefix probabilities at position *k* to position *k*–1, where prefix probabilities are the sum of tree probabilities compatible with morphemes until position *k* (Earley, 1970; Stolcke, 1995). PCFGs are hierarchical models, which should accurately predict not only local dependencies of linear words (e.g. *digitally*), but also non-local dependencies of nested words (e.g. *unpredictable*).

2.5 Evaluation metrics

The information-theoretic complexity metric, *surprisal*, was employed as linking hypothesis that bridges the gap between representation and processing (Hale, 2001; Levy, 2008). Surprisal of morpheme *m*, $I(m)$, is defined as Equation (1):

$$I(m) = \log_2 \frac{1}{P(m)} = -\log_2 P(m) \qquad (1)$$

where $P(m)$ is the probability of morpheme m computed by computational models via respective incremental algorithms. Surprisal was originally proposed to explain behavioral measures such as reading times in self-paced reading experiments and fixation durations in eye-tracking experiments (Boston et al., 2008; Demberg and Keller, 2008; Roark et al., 2009; Frank and Bod, 2011; Fossum and Levy, 2012). Recently, surprisal has also been extended to neural measures like N400 components in EEG experiments and BOLD signals in fMRI experiments (Frank et al., 2015; Brennan et al., 2016; Willems et al., 2016; Henderson et al., 2016; Nelson et al., 2017; Lopopolo et al., 2017).

Assuming further that morphological processing is incremental (cf. prefix stripping; Taft and Forster, 1975; Stockall et al., 2019), we compute surprisal of morphologically complex words as *cumulative surprisal*, the cumulative sum of surprisal of component morphemes. Cumulative surprisal of word *w*, $I(w)$, is defined as Equation (2):

$$I(w) = I(m_1, ..., m_n) = \sum_{i=1}^{n} I(m_i) \qquad (2)$$

where $I(m)$ is the surprisal of morpheme m computed by computational models.

Two evaluation metrics are then derived from cumulative surprisal: neurological and error accuracies (cf. Frank et al., 2015; Sprouse et al., 2018). The neurological accuracy of model *M*, $NA(M)$, is defined as Equation (3):

$$NA(M) = D_B - D_M \qquad (3)$$

where D_B and D_M are deviance defined as –2 times log-likelihoods of baseline and target models, respectively. Neurological accuracy quantifies decreases in deviance ($-\Delta D$) and evaluates how well computational models explain human neural activities beyond control predictors included in the baseline model (cf. Frank et al., 2015).

The error accuracy of model *M*, $EA(M)$, is defined as Equation (4):

$$EA(M) = \sum_{i=1}^{n} |\epsilon_B(w_i)| - |\epsilon_M(w_i)| \qquad (4)$$

where $\epsilon_B(w)$ and $\epsilon_M(w)$ are residual errors of baseline and target models for word w, respectively. Error accuracy quantifies decreases in absolute residual errors ($-\Delta|\epsilon|$) and evaluates cost-benefit tradeoffs of computational models (cf. Sprouse et al., 2018). We compute error accuracies of computational models with respect to linear and nested morphological structures to address the question whether hierarchical models make better predictions for nested words than linear models.

2.6 Statistical analyses

We performed linear mixed-effects regression (Baayen et al., 2008) by averaging neural activities within the functionally defined region of interest (fROI) based on spatiotemporal cluster permutation regression (Maris and Oostenveld, 2007). In the previous literature (cf. Gwilliams et al., 2016), *lemma frequency* has been proposed as a significant predictor of the M170 and, thus, employed as the predictor of interest for spatiotemporal regression. Lemma frequency (cf. del Prado Martin et al., 2004) is defined as the sum of frequencies of words that share the same lemma. For example, the lemma frequency of *globalization* is the sum of frequencies of *globe*, *global*, *globalize*, and so on. Spatiotemporal regression in the left inferior temporal lobe and the 150-200 time window with log-transformed lemma frequency as target predictor and squared length as control predictor identified the significant cluster where the clear M170 peak can be observed, as shown in Figure 1. Finally, the neural activities were averaged over space and time within the fROI to compute by-trial dSPMs (Dale et al., 2000), which were then exported to R for mixed-effects regression.

Linear mixed-effects regression was implemented with `lme4` package (Bates et al., 2015) in R. The baseline regression model was first fitted with by-trial dSPMs as the dependent variable, control predictors as fixed effects, and by-subject and by-word random intercepts as random effects. For each computational model, the target regression model was then fitted with cumulative surprisal included as an additional fixed effect on top

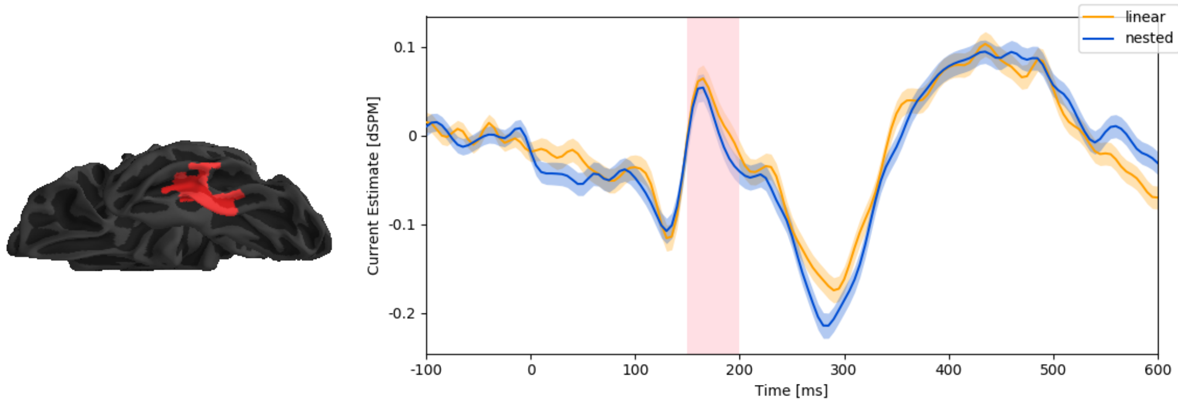


Figure 1: fROI for linear mixed-effects regression. Left: spatial extent defined as the significant cluster identified via spatiotemporal regression in the left inferior temporal lobe and the 150-200 time window with log-transformed lemma frequency as target predictor and squared length as control predictor; Right: temporal extent averaged over the significant cluster and categorized by linear and nested morphological structures. The *x*-axis is time in milliseconds, while the *y*-axis is neural activities in dSPM (Dale et al., 2000). Color indicates two morphological structures: yellow = linear, blue = nested. Pink vertical span marks the 150-200 ms time window.

of control predictors and random effects held constant. The control predictor was squared length (New et al., 2006) also included to functionally define the ROI. Mixed-effects models were fitted via Maximum Likelihood Estimation with `nlminb` optimizer in `optimx` package and the maximum number of iterations R permits. Given that the baseline and target models are minimally different only in cumulative surprisal, computational models can be evaluated with nested model comparisons via log-likelihood ratio tests based on χ^2-distribution with $df = 1$, where df is the difference in number of parameters between nested models.

3 Results

3.1 Neurological accuracy

Neurological accuracies of computational models are summarized in Figure 2, where the *x*-axis is computational models and the *y*-axis is neurological accuracies (i.e. decreases in deviance). The horizontal dashed line is $\chi^2 = 3.84$, the critical χ^2-statistic at $p = 0.05$ with $df = 1$.

Nested model comparisons via log-likelihood ratio tests revealed that while no "amorphous" models were statistically significant, all "morphous" models were statistically significant ($p < 0.01$). Among those "morphous" models, PCFG was most neurologically accurate: PCFG ($\chi^2 = 8.48$, $p < 0.01$) > Markov Model ($\chi^2 = 8.15$, $p < 0.01$) > HMM ($\chi^2 = 6.92$, $p < 0.01$) > Character ($\chi^2 = 0.19$, *ns*) > Syllable ($\chi^2 = 0.02$, *ns*).

3.2 Error accuracy

Error accuracies of computational models are summarized in Figure 3, where the *x*-axis is computational models and the *y*-axis is error accuracies (i.e. decreases in absolute residual errors), categorized into linear and nested morphological structures and averaged across individual derivational affixes. The horizontal dashed line indicates a "tie" borderline where computational models do not diverge from the baseline model. More positive and negative error accuracies mean better and worse predictions relative to the baseline model.

For linear words, all neurologically accurate "morphous" models made significant contributions, among which Markov Model made best predictions relative to the baseline model. For nested words, interestingly, PCFG was the only computational model which reduced residual errors, while linear models such as HMM and Markov Model made only slight or even worse predictions relative to the baseline model, respectively.

4 Discussion

In summary, our MEG experiment and computational modeling demonstrated that "morphous" models of morphological processing outperformed "amorphous" models and, importantly, PCFG was most neurologically accurate among those "morphous" models. We can conclude from these results that morphemes are neurologically represented in the human brain (pace Baayen

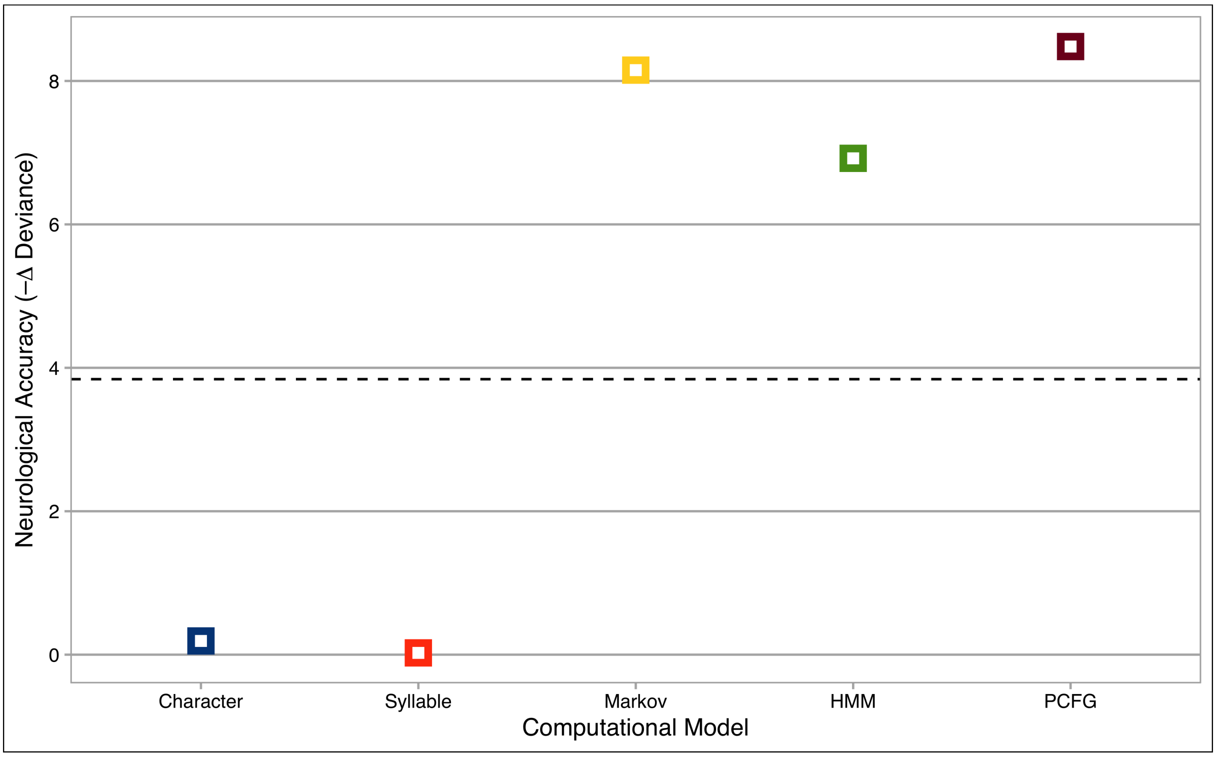


Figure 2: Neurological accuracies of computational models. The x-axis is computational models, while the y-axis is neurological accuracies (i.e. decreases in deviance). Points represent computational models: blue = Character Markov Model, orange = Syllable Markov Model, yellow = Morpheme Markov Model, green = Hidden Markov Model, purple = Probabilistic Context-Free Grammar. The horizontal dashed line is $\chi^2 = 3.84$, the critical χ^2-statistic at $p = 0.05$ with $df = 1$. All "morphous" models were statistically significant ($p < 0.01$).

et al., 2011; Milin et al., 2017) and parsed into hierarchical morphological structures (pace Virpioja et al., 2017; Hakala et al., 2018). In addition, this paper successfully generalized the computational modeling technique developed in the sentence processing literature (Frank et al., 2015; Brennan et al., 2016) to morphological processing.

Moreover, error accuracies of computational models indicated that PCFG better explained nested words with non-local dependencies between prefixes and suffixes than linear models such as Markov Model and HMM. This result follows straightforwardly from formal language theory, where linear and nested words are finite-state and context-free languages in the Chomsky hierarchy (Hopcroft and Ullman, 1979; Partee et al., 1990; Sipser, 1997), the former of which can be modeled by both linear and hierarchical models, but the latter of which can only be parsed by hierarchical models like PCFG. Furthermore, from the probabilistic perspective, linear models have trouble with transition probabilities from prefixes to roots in nested words (e.g. *unpredictable*) because prefixes (e.g. *un-*) and roots (e.g. *predict*) form no morphological constitutes (e.g. **unpredict*) and thus never appear in the training data.

Now the theoretical question arises why low-level visual evoked response components like M170 in the visual ventral stream "know" high-level linguistic representations like abstract hierarchical structures. One possibility is that, given the functional connectivity between the left fusiform gyrus and the left inferior frontal gyrus in visual word recognition (Pammer et al., 2004), M170 can be modulated in a top-down feedback manner by "Broca's area", the traditional "language" area proposed to process abstract hierarchical structures (Friederici, 2002, 2012). This possibility becomes even less surprising if visual cortex can be sensitive to abstract hierarchical structures (Dikker et al., 2009). Therefore, the functional connectivity between the left fusiform and inferior frontal gyri remains to be empirically investigated in the future research (Carreiras et al., 2014; Woodhead et al., 2014).

Nevertheless, there are several limitations with our computational modeling. One of the several important issues is that "amorphous" models in-

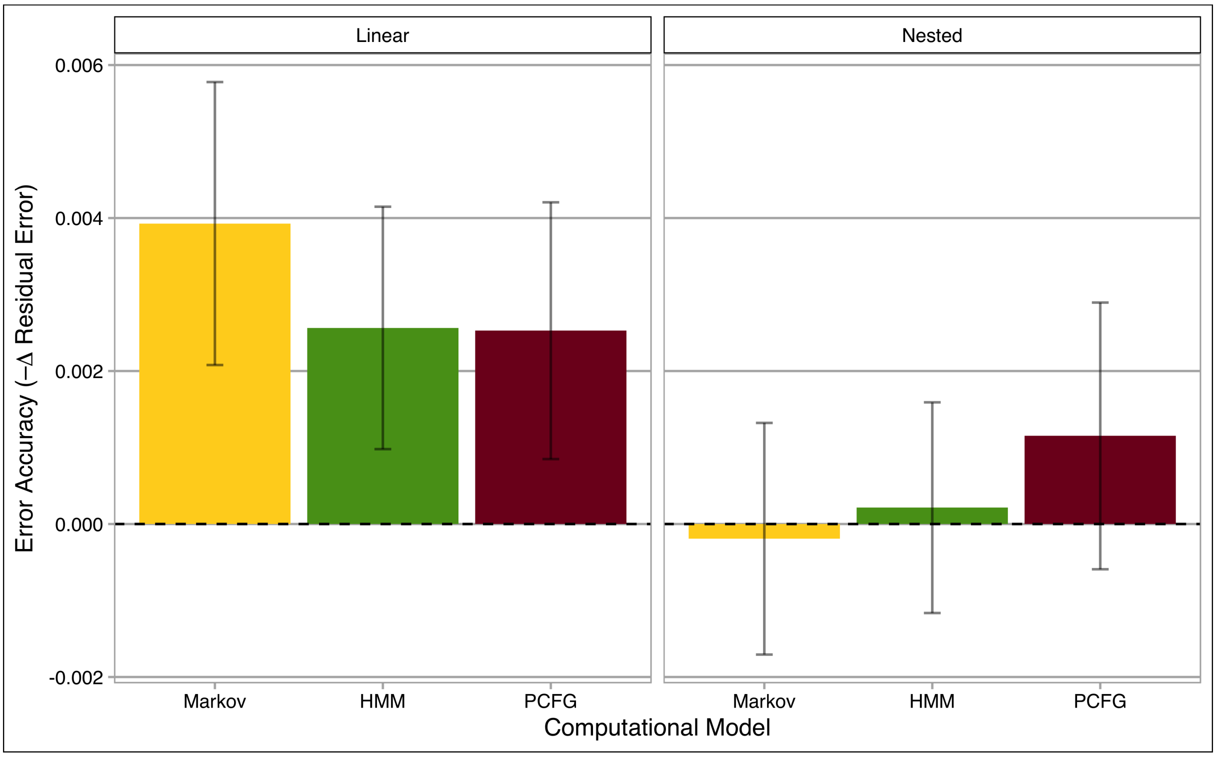


Figure 3: Error accuracies of computational models. The *x*-axis is computational models, while the *y*-axis is error accuracies (i.e. decreases in absolute residual errors), categorized into linear (Left) and nested (Right) morphological structures and averaged across individual derivational affixes. The horizontal dashed line indicates a "tie" borderline where computational models do not diverge from the baseline model, and more positive and negative error accuracies mean better and worse predictions relative to the baseline model.

vestigated in this paper are too simplistic as compared to computational models recently proposed in the morphological processing literature such as *Naive Discriminative Learning* (Baayen et al., 2011; Milin et al., 2017) or *Linear Discriminative Learning* (Baayen et al., 2018, 2019). Those state-of-the-art computational models of morphological processing remain to be constructed and evaluated against human neural activities and computational models investigated in this paper.

5 Conclusion

In this paper, we conducted a magnetoencephalography (MEG) experiment where participants performed visual lexical decision on morphologically complex words and, generalizing the computational modeling technique developed in the sentence processing literature (Frank et al., 2015; Brennan et al., 2016), computationally modeled morphological processing in the human brain, with special focus on the VWFA in the visual ventral stream. Five neuro-computational models of morphological processing were constructed and evaluated against human neural activities in order to investigate whether morphemes are neurologically represented in the human brain and parsed into hierarchical morphological structures: Character Markov Model and Syllable Markov Model as "amorphous" models without morpheme units, and Morpheme Markov Model, Hidden Markov Model (HMM), and Probabilistic Context-Free Grammar (PCFG) as "morphous" models with morpheme units structured linearly or hierarchically. Our MEG experiment and computational modeling demonstrated that "morphous" models of morphological processing outperformed "amorphous" models, PCFG was most neurologically accurate among those "morphous" models, and PCFG better explained nested words with non-local dependencies between prefixes and suffixes. These results strongly suggest that morphemes are neurologically represented in the human brain and parsed into hierarchical morphological structures. In conclusion, neuro-computational modeling of natural language must be a promising future direction in the cognitive computational neuroscience of language (Kriegeskorte and Douglas, 2018; Naselaris et al., 2018).

Acknowledgments

We would like to thank three anonymous reviewers of the *Society for Computation in Linguistics* and the members of the Neuroscience of Language Lab at New York University for valuable comments and suggestions. This work was supported by JSPS KAKENHI Grant Numbers JP18H05589 and JP19H04990 (YO) and the NYU Abu Dhabi Institute Grant Number G1001 (AM).

References

Harald Baayen, Yu-Ying Chuang, , and James Blevins. 2018. Inflectional morphology with linear mappings. *The Mental Lexicon*, 13:232–270.

Harald Baayen, Douglas Davidson, and Douglas Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59:390–412.

Harald Baayen, Petar Milin, Dusica Filipovic Durdevic, Peter Hendrix, and Marco Marelli. 2011. An Amorphous Model for Morphological Processing in Visual Comprehension Based on Naive Discriminative Learning. *Psychological Review*, 118:438–481.

Harald Baayen, Yu-Ying Chuang andElnaz Shafaei-Bajestan, and James Blevins. 2019. The Discriminative Lexicon: A Unified Computational Model for the Lexicon and Lexical Processing in Comprehension and Production Grounded Not in (De)Composition but in Linear Discriminative Learning. *Complexity*, page Article 4895891.

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Douglas Bates, Martin Maechler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67:1–48.

Laurie Bauer. 2014. Grammaticality, acceptability, possible words and large corpora. *Morphology*, 24:83–103.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Marisa Boston, John Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. 2008. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2:1–12.

Jonathan Brennan, Edward Stabler, Sarah Van Wagenen, Wen-Ming Luh, and John Hale. 2016. Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and Language*, 157-158:81–94.

Manuel Carreiras, Blair Armstrong, Manuel Perea, and Ram Frost. 2014. The what, when, where, and how of visual word recognition. *Trends in Cognitive Sciences*, 18:90–98.

Laurent Cohen, Stanislas Dehaene, Lionel Naccache, Stephane Lehericy, Ghislaine Dehaene-Lambertz, Marie-Anne Henaff, and Francois Michel. 2000. The visual word form area: Spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. *Brain*, 123:291–307.

Laurent Cohen, Stephane Lehericy, Florence Chochon, Cathy Lemer, Sohie Rivaud, and Stanislas Dehaene. 2002. Language-specific tuning of visual cortex? Functional properties of the Visual Word Form Area. *Brain*, 125:1054–1069.

Mathias Creutz and Crista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4:3.

Anders M. Dale, Arthur K. Liu, Bruce R. Fischl, Randy L. Buckner, John W. Belliveau, Jeffrey D. Lewine, and Eric Halgren. 2000. Dynamic Statistical Parametric Mapping: Combining fMRI and MEG for High-Resolution Imaging of Cortical Activity. *Neuron*, 26:55–67.

Stanislas Dehaene, Gurvan Le Clec'H, Jean-Baptiste Poline, Denis Le Bihan, and Laurent Cohen. 2002. The visual word form area: A prelexical representation of visual words in the fusiform gyrus. *NeuroReport*, 13:321–325.

Stanislas Dehaene, Laurent Cohen, Mariano Sigman, and Fabien Vinckier. 2005. The neural code for written words: a proposal. *Trends in Cognitive Sciences*, 9:335–341.

Stanislas Dehaene, Lionel Naccache, Laurent Cohen, Denis Le Bihan, Jean-Francois Mangin, Jean-Baptiste Poline, and Denis Riviere. 2001. Cerebral mechanisms of word masking and unconscious repetition priming. *Nature NeuroScience*, 4:752–758.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109:193–210.

Suzanne Dikker, Hugh Rabagliati, and Liina Pylkkänen. 2009. Sensitivity to syntax in visual cortex. *Cognition*, 110:293–321.

Jay Earley. 1970. An ecient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 13:94–102.

Victoria Fossum and Roger Levy. 2012. Sequential vs. hierarchical syntactic models of human incremental sentence processing. *Proceedings of the*

3rd Workshop on Cognitive Modeling and Computational Linguistics, pages 61–69.

Stefan Frank and Rens Bod. 2011. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22:829–834.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.

Angela Friederici. 2002. Towards a neural basis of auditory sentence processing. *Trends in Cognitive Sciences*, 6:78–84.

Angela Friederici. 2012. The cortical language circuit: from auditory perception to sentence comprehension. *Trends in Cognitive Sciences*, 16:262–268.

Joseph Fruchter and Alec Marantz. 2015. Decomposition, lookup, and recombination: MEG evidence for the Full Decomposition model of complex visual work recognition. *Brain and Language*, 143:81–96.

Joseph Fruchter, Linnaea Stockall, and Alec Marantz. 2013. MEG masked priming evidence for form-based decomposition of irregular verbs. *Frontiers in Human Neuroscience*, 7:798.

Laura Gwilliams, Gwyneth Lewis, and Alec Marantz. 2016. Functional characterisation of letter-specific responses in time, space and current polarity using magnetoencephalography. *Neuroimage*, 132:320–333.

Tero Hakala, Annika Hulten, Minna Lehtonen, Krista Lagus, and Riitta Salmelin. 2018. Information properties of morphologically complex words modulate brain activity during word reading. *Human Brain Mapping*, 39:2583–2595.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. *Proceedings of NAACL-2001*, pages 159–166.

Morris Halle. 1973. Prolegomena to a Theory of Word Formation. *Linguistic Inquiry*, 4:3–16.

Jennifer Hay. 2003. *Causes and Consequences of Word Structure*. Routledge, New York, NY.

P. Helenius, A. Tarkiainen, P. Cornelissen, P.C. Hansen, and R. Salmelin. 1999. Dissociation of Normal Feature Analysis and Deficient Processing of Letter-strings in Dyslexic Adults. *Cerebral Cortex*, 9:476–483.

John M. Henderson, Wonil Choi, Matthew W. Lowder, and Fernanda Ferreira. 2016. Language structure in the brain: A fixation-related fMRI study of syntactic surprisal in reading. *NeuroImage*, 132:291–300.

Gregory Hickok and David Poeppel. 2007. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8:393–402.

John Hopcroft and Jeffrey Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA.

Nikolaus Kriegeskorte and Pamela Douglas. 2018. Cognitive computational neuroscience. *Nature Neuroscience*, 21:1148–1160.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.

Gwyneth Lewis, Olla Solomyak, and Alec Marantz. 2011. The Neural Basis of Obligatory Decomposition of Suffixed Words. *Brain and Language*, 118:118–127.

Gary Libben. 2003. Morphological parsing and morphological structure. In Egbert Assink and Dominiek Sandra, editors, *Reading Complex Words*, pages 221–239. Kluwer, New York.

Gary Libben. 2006. Getting at psychological reality: On- and off-line tasks in the investigation of hierarchical morphological structure. In G. Wiebe, G. Libben, T. Priestly, R. Smyth, and S. Wang, editors, *Phonology, Morphology, and the Empirical Imperative*, pages 349–369. Crane, Taipei.

Alessandro Lopopolo, Stefan L. Frank, Antal van den Bosch, and Roel M. Willems. 2017. Using stochastic language models (SLM) to map lexical, syntactic, and phonological information processing in the brain. *PLoS ONE*, 12:e0177794.

Eric Maris and Robert Oostenveld. 2007. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, 164:177–190.

Petar Milin, Laurie Feldman, Michael Ramscar, Peter Hendrix, and Harald Baayen. 2017. Discrimination in lexical decision. *PLoS ONE*, 12.

Thomas Naselaris, Danielle Bassett, Alyson Fletcher, Konrad Kording, Nikolaus Kriegeskorte, Hendrikje Nienborg, Russell Poldrack, Daphna Shohamy, and Kendrick Kay. 2018. Cognitive Computational Neuroscience: A New Conference for an Emerging Discipline. *Trends in Cognitive Sciences*, 22:365–367.

Matthew J Nelson, Imen El Karoui, Kristof Giber, Xiaofang Yang, Laurent Cohen, Hilda Koopman, Sydney S Cash, Lionel Naccache, John T Hale, Christophe Pallier, et al. 2017. Neurophysiological dynamics of phrase-structure building during sentence processing. *Proceedings of the National Academy of Sciences*, 114:3669–3678.

Boris New, Ludovic Ferrand, Christophe Pallier, and Marc Brysbaert. 2006. Reexamining the word length effect in visual word recognition: New evidence from the English Lexicon Project. *Psychonomic Bulletion and Review*, 13:45–52.

Carolus Oldfield. 1971. The assessment and analysis of handedness: The Edinburgh inventory. *Neuropsychologia*, 9:97–113.

Yohei Oseki, Charles Yang, and Alec Marantz. 2019. Modeling Hierarchical Syntactic Structures in Morphological Processing. *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 43–52.

Kristen Pammer, Peter Hansen, Morten Kringelbach, Ian Holliday, Gareth Barnes, Arjan Hillebrand, Krish Singh, and Piers Cornelissen. 2004. Visual word recognition: the first half second. *NeuroImage*, 22:1819–1825.

Barbara Partee, Alice ter Meulen, and Robert Wall. 1990. *Mathematical Methods in Linguistics*. Springer, Dordrecht.

Jonathan Peirce. 2007. PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162:8–13.

Jonathan Peirce. 2009. Generating stimuli for neuroscience using PsychoPy. *Frontiers in Neuroinformatics*, 2:10.

Ingo Plag and Harald Baayen. 2009. Suffix Ordering and Morphological Processing. *Language*, 85:109–152.

Fermin Moscoso del Prado Martin, Aleksandar Kostic, and Harald Baayen. 2004. Putting the bits together: an information theoretical perspective on morphological processing. *Cognition*, 94:1–18.

Liina Pylkkänen and Alec Marantz. 2003. Tracking the time course of word recognition with MEG. *Trends in Cognitive Sciences*, 7:187–189.

Lawrence Rabinar. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77:257–286.

Brian Roark, Asaf Bachrach, Carlos Cardenas, and Christophe Pallier. 2009. Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 324–333.

Michael Sipser. 1997. *Introduction to the Theory of Computation*. PWS Publishing, Boston, MA.

Olla Solomyak and Alec Marantz. 2010. Evidence for Early Morphological Decomposition in Visual Word Recognition: A Single-Trial Correlational MEG Study. *Journal of Cognitive Neuroscience*, 22:2042–2057.

Yoonsang Song, Youngah Do, Jongbong Lee, Arthur Thompson, and Eileen Waegemaekers. 2019. The reality of hierarchical morphological structure in multimorphemic words. *Cognition*, 183:269–276.

Jon Sprouse, Sagar Indurkhya, Beracah Yankama, Sandiway Fong, and Robert C. Berwick. 2018. Colorless green ideas do sleep furiously: gradient acceptability and the nature of the grammar. *Linguistic Review*, 35:575–599.

Linnaea Stockall, Christina Manouilidou, Laura Gwilliams, Kyriaki Neophytou, and Alec Marantz. 2019. Prefix Stripping Re-Re-Revisited: MEG Investigations of Morphological Decomposition and Recomposition. *Frontiers in Psychology*, 10:1964.

Andreas Stolcke. 1995. An ecient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21:165–201.

M. Taft. 1979. Recognition of affixed words and the word frequency effect. *Memory and Cognition*, 7:263–272.

M. Taft. 2004. Morphological decomposition and the reverse base frequency effect. *The Quarterly Journal of Experimental Psychology*, 57:745–765.

M. Taft and K. I. Forster. 1975. Lexical storage and retrieval of prefixed words. *Journal of Verbal Learning and Verbal Behavior*, 14:638–647.

A. Tarkiainen, P. Helenius, P. C. Hansen, P. L. Cornelissen, and R Salmelin. 1999. Dynamics of letter string perception in the human occipitotemporal cortex. *Brain*, 122:2119–2132.

Sami Virpioja, Minna Lehtonen, Annika Hulten, Henna Kivikari, Riitta Salmelin, and Krista Lagus. 2017. Using Statistical Models of Morphology in the Search for Optimal Units of Representation in the Human Mental Lexicon. *Cognitive Science*, pages 1–35.

Roel M. Willems, Stefan L. Frank, Annabel D. Nijhof, Peter Hagoort, and Antal van den Bosch. 2016. Prediction During Natural Language Comprehension. *Cerebral Cortex*, 26:2506–2516.

Z.V.J. Woodhead, G.R. Barnes, W. Penny, R. Moran, S. Teki, C.J. Price, and A.P. Leff. 2014. Reading Front to Back: MEG Evidence for Early Feedback Effects During Word Recognition. *Cerebral Cortex*, 24:817–825.

Linda Zirkel. 2010. Prefix combinations in English: structural and processing factors. *Morphology*, 20:239–266.

Etyan Zweig and Liina Pylkkänen. 2009. A Visual M170 Effect of Morphological Complexity. *Language and Cognitive Processes*, 24:412–439.

The lexical and grammatical sources of neg-raising inferences

Hannah Youngeun An
Department of Computer Science
University of Rochester

Aaron Steven White
Department of Linguistics
University of Rochester

Abstract

We investigate *neg(ation)-raising* inferences, wherein negation on a predicate can be interpreted as though in that predicate's subordinate clause. To do this, we collect a large-scale dataset of neg-raising judgments for effectively all English clause-embedding verbs and develop a model to jointly induce the semantic types of verbs and their subordinate clauses and the relationship of these types to neg-raising inferences. We find that some neg-raising inferences are attributable to properties of particular predicates, while others are attributable to subordinate clause structure.

1 Introduction

Inferences that are triggered (at least in part) by particular lexical items provide a rich test bed for distinguishing the relative semantic contribution of lexical items and functional structure. One class of such inferences that has garnered extended attention is *neg(ation)-raising*, wherein negation on a predicate can be interpreted as though in that predicate's subordinate clause (Fillmore, 1963; Bartsch, 1973; Horn, 1978; Gajewski, 2007). For example, a neg-raising inference is triggered by (1) while one is not triggered by (2).

(1) Jo doesn't think that Bo left.
$\rightsquigarrow$Jo thinks that Bo didn't leave.

(2) Jo doesn't know that Bo left.
$\not\rightsquigarrow$Jo knows that Bo didn't leave.

Though accounts vary with respect to whether neg-raising inferences are explained as a syntactic or a pragmatic phenomenon, all associate these inferences with particular predicates in some way or other—e.g. *think, believe, suppose, imagine, want*, and *expect* are often taken to be associated with neg-raising inferences as a matter of knowledge one has about those predicates, while *say, claim, regret*, and *realize* are not (Horn, 1971, 1978).

One challenge for such approaches is that whether a neg-raising inference is triggered varies with aspects of the context, such as the predicate's subject—e.g. (3a) triggers the inference that the speaker thinks Jo didn't leave—and tense—e.g. (3b) does not trigger the same inference as (3a).

(3) a. I don't know that Jo left.
b. I didn't know that Jo left.

While some kinds of variability can be captured by standing accounts, other kinds have yet to be discussed at all. For example, beyond a predicate's subject and tense, the syntactic structure of its clausal complement also appears to matter: (4a) and (5a) can both trigger neg-raising interpretations, while (4b) and (5b) cannot.

(4) a. Jo wasn't thought to be very intelligent.
b. Jo didn't think to get groceries.

(5) a. Jo wasn't known to be very intelligent.
b. Jo didn't know to get groceries.

Should these facts be chalked up to properties of the predicates in question? Or are they general to how these predicates compose with their complements? These questions are currently difficult to answer for two reasons: (i) there are no existing, lexicon-scale datasets that measure neg-raising across a variety of contexts—e.g. manipulating subject, tense and complement type; and (ii) even if there were, no models currently exist for answering these questions given such a dataset.

We fill this lacuna by (i) collecting a large-scale dataset of neg-raising judgments for effectively all English clause-embedding verbs with a variety of both finite and non-finite complement types; and (ii) extending White and Rawlins' (2016) model of s(emantic)-selection, which induces semantic type signatures from syntactic distribution, with a module that associates semantic types with the inferences they trigger. We use this model to jointly induce semantic types and their relationship to neg-raising inferences, showing that the best fitting model attributes some neg-raising inferences to properties of particular predicates and others to general properties of syntactic structures.[1]

We begin with background on theoretical approaches to neg-raising, contrasting the two main types of accounts: syntactic and pragmatic (§2). We then present our methodology for measuring neg-raising across a variety of predicates and syntactic contexts (§3) as well as our extension of White and Rawlins' s-selection model (§4). Finally, we discuss the results of fitting (§5) our model to our neg-raising dataset (§6).

2 Background

Two main types of approaches have been proposed to account for neg-raising interpretations: syntactic and pragmatic (see Zeijlstra 2018; Crowley 2019 for reviews). We do not attempt to adjudicate between the two here—rather aiming to establish the explanatory devices available to each for later interpretation relative to our modeling results.

Syntactic Approach In syntactic approaches, neg-raising interpretations arise from some syntactic relation between a matrix negation and an unpronounced embedded negation that is licensed by the neg-raising predicate. This is classically explained via a syntactic rule that "raises" the negation from the subordinate clause to the main clause, as in (6), though accounts using alternative syntactic relations exist (Fillmore 1963; Kiparsky 1970; Jackendoff 1971; Pollack 1976; Collins and Postal 2014, 2017, 2018; cf. Klima, 1964; Zeijlstra, 2018; see also Lasnik, 1972).

(6) Jo does not believe Bo did ___ leave.

Evidence for syntactic accounts comes from the distribution of negative polarity items, Horn-clauses, and island phenomena (Horn, 1971; Collins and Postal, 2014, 2017, 2018; cf. Zwarts, 1998; Gajewski, 2011; Chierchia, 2013; Horn, 2014; Romoli and Mandelkern, 2019).

Purely syntactic approaches to neg-raising have effectively one method for explaining variability in neg-raising inferences relative to subject, tense, and subordinate clause structure (as discussed in §1): if a certain lexical item—e.g. *know*—occurs in some sentence that licenses a neg-raising inference—e.g. (5a)—and another that doesn't—e.g. (5b)—one must say that the structure in the first differs from the second in such a way that the first allows the relevant syntactic relation while the second does not. This implies that, even in cases like (3a) v. (3b), where there is no apparent structural difference (beyond the subject), the structures differ on some neg-raising-relevant property. This can be implemented by saying that, e.g. the same verb can select for two different structural properties—one that licenses neg-raising and one that does not—or that the verb is somehow ambiguous and its variants differ with respect to some neg-raising-relevant, syntactic property.

Semantic/Pragmatic Approach In semantic/pragmatic approaches, neg-raising interpretations are derived from an *excluded middle* (EM or *opinionatedness*) inference (Bartsch, 1973; Horn, 1978; Horn and Bayer, 1984; Tovena, 2001; Gajewski, 2007; Romoli, 2013; Xiang, 2013; Homer, 2015). This approach posits that, anytime a neg-raising predicate v is used to relate entity x with proposition p, the hearer assumes that either $x\ v\ p$ or $x\ v\ \neg p$. For example, in the case of *believe*, as in (7), the hearer would assume that Jo either believes that Bo left or that Bo didn't leave.

(7) Jo believes that Bo left.
  a. *truth conditions:* x BELIEVE p
  b. *inference:* x BELIEVE $p \vee x$ BELIEVE $\neg p$

The EM inference is impotent in the positive cases but drives further inferences in the negative, where the first EM disjunct is cancelled by the truth conditions: if Jo doesn't believe that Bo left and Jo believes that Bo left or that Bo didn't leave, then Jo must believe that Bo didn't leave.

(8) Jo doesn't believe that Bo left.
  a. *truth conditions:* x ¬ BELIEVE p
  b. *inference:* $\cancel{x \text{ BELIEVE } p \vee}$ x BELIEVE $\neg p$

To capture non-neg-raising predicates, one must then say that some predicates trigger the EM inference, while others don't (Horn, 1989). However, such lexical restrictions alone cannot exhaustively explain the variability in whether verbs trigger presuppositions with certain subjects, as noted for (2) and (3a). To explain this, Gajewski (2007) posits that neg-raising predicates are soft presupposition triggers. Effectively, the EM inferences are defeasible, and when they are cancelled, the neg-raising inference does not go through (Abusch, 2002). This is supported by cases of explicit cancella-

[1] Data are available at megaattitude.io.

tion of the EM inference—e.g. the neg-raising inference (9c) that would otherwise be triggered by (9b) does not go through in the context of (9a).

(9) a. Bill doesn't know who killed Caesar. He isn't even sure whether or not Brutus and Caesar lived at the same time. So...
b. Bill doesn't believe Brutus killed Caesar.
c. $\not\rightsquigarrow$ Bill believes Brutus didn't kill Caesar.

This sort of explanation relies heavily on semantic properties of particular verbs and naturally covers variability that correlates with subject and tense differences—e.g. (3a) v. (3b)—since facts about how one discusses their own belief or desire states, in contrast to others belief states, at different times plausibly matter to whether a hearer would make the EM inference. The explanation for variation relative to subordinate clause structure is less clear but roughly two routes are possible: (i) some property of the subordinate clause licenses (or blocks) EM inferences; and/or (ii) predicate ambiguity correlates with which subordinate clause structure (or property thereof) a predicate selects.

Abstracting the Approaches Across both approaches, there are roughly three kinds of explanations for neg-raising inferences that can be mixed-and-matched: (i) lexical properties might directly or indirectly (e.g. via an EM inference) license a neg-raising inference; (ii) properties of a subordinate clause structure might directly or indirectly license a neg-raising inference; and/or (iii) lexical and structural properties might interact—e.g. via selection—to directly or indirectly license a neg-raising inference. We incorporate these three kinds of explanation into our models (§4), which we fit to the data described in the next section.

3 Data

We develop a method for measuring neg-raising analogous to White and Rawlins-White et al.'s (2018) method for measuring veridicality inferences. With the aim of capturing the range of variability in neg-raising inferences across the lexicon, we deploy this method to test effectively all English clause-embedding verbs in a variety of subordinate clause types—finite and nonfinite—as well as matrix tenses—*past* and *present*—and matrix subjects—*first* and *third person*.

Method Participants are asked to answer questions like (10) using a 0-1 slider, wherein the first italicized sentence has negation in the matrix clause and the second italicized sentence has negation in the subordinate.[2]

(10) If I were to say *I don't think that a particular thing happened*, how likely is it that I mean *I think that that thing didn't happen*?

Because some sentences, such the italicized in (11), sound odd with negation in the matrix clause, participants are asked to answer how easy it is to imagine someone actually saying the sentence—again, on a 0-1 slider. The idea here is that the harder it is for participants to imagine hearing a sentence, the less certain they probably are about the judgment to questions like (10).

(11) How easy is it for you to imagine someone saying *I don't announce that a particular thing happened*?

Acknowledging the abuse of terminology, we refer to responses to (11) as *acceptability responses*. We incorporate these responses into our model (§4) as weights determining how much to pay attention to the corresponding *neg-raising response*.

Materials We use the MegaAcceptability dataset of White and Rawlins (2016) as a basis on which to construct acceptable items for our experiment. MegaAcceptability contains ordinal acceptability judgments for 50,000 sentences, including 1,000 clause-embedding English verbs in 50 different syntactic frames. To avoid typicality effects, these frames are constructed to contain as little lexical content as possible besides the verb at hand—a method we follow here. This is done by ensuring that all NP arguments are indefinite pronouns *someone* or *something* and all verbs besides the one being tested are *do*, *have* or *happen*. We focus on the six frames in (12)–(17).

(12) [NP _ that S]
Someone <u>knew</u> that something happened.

(13) [NP _ to VP[+EV]]
Someone <u>liked</u> to do something.

(14) [NP _ to VP[-EV]]
Someone <u>wanted</u> to have something.

(15) [NP be _ that S]
Someone was <u>told</u> that something happened.

(16) [NP be _ to VP[+EV]]
Someone was <u>ordered</u> to do something.

(17) [NP be _ to VP[-EV]]
Someone was <u>believed</u> to have something.

[2]The full task instructions are given in Appendix A.

These frames were chosen so as to manipulate (i) the presence and absence of tense in the subordinate clause; (ii) the presence or absence of a direct object; and (iii) the lexical aspect of the complement. The frames with direct objects were presented in passivized form so that they were acceptable with both communicative predicates—e.g. *tell*—and emotive predicates—e.g. *sadden*—the latter of which tend to occur with expletive subjects. Lexical aspect was manipulated because some verbs—e.g. *believe*—are more acceptable with nonfinite subordinate clauses headed by a stative than ones headed by an eventive, while others—e.g. *order*—show the opposite pattern.

In light of the variability in neg-raising inferences across the same verb in different tenses—compare again (3a) and (3b)—we aim to manipulate the matrix tense of each clause-taking verb in our experiment. This is problematic, because the MegaAcceptability dataset only contains items in the past tense. We could simply manipulate the tense for any acceptable sentences based on such past tense items, but some verbs do not sound natural in the present tense with some subordinate clauses—compare the sentences in (18).

(18) a. Jo wasn't told that Mary left.
b. Jo isn't told that Mary left.

To remedy this, we extend MegaAcceptability with tense/aspect information by collecting acceptability judgments for modified versions of each sentence in MegaAcceptability, where the target verb is placed in either present or past progressive.[3] Combined with MegaAcceptability, our extended dataset results in a total of 75,000 verb-tense-frame pairs: 50,000 from the MegaAcceptability dataset and 25,000 from our dataset. From this combined dataset, we take past and present tense items rated on average 4 out of 7 or better (after rating normalization), for our experiment. This yields 3,968 verb-tense-frame pairs and 925 unique verbs. With our subject manipulation (first v. third person), the number of items doubles, producing 7,936 items. Table 1 summarizes the distribution of verbs in each frame and tense.

To construct items, we follow the method of White et al. (2018) of "bleaching" all lexical category words in our sentences (besides the subordinate clause-taking verb) by realizing NPs as *a particular person* or *a particular thing*. Verbs are

[3]See Appendix B for details.

Matrix tense	Frame	# verbs
past	NP _ that S	556
	NP _ to VP[+EV]	400
	NP _ to VP[-EV]	359
	NP be _ that S	255
	NP be _ to VP[+EV]	461
	NP be _ to VP[-EV]	460
present	NP _ that S	413
	NP _ to VP[+EV]	219
	NP _ to VP[-EV]	155
	NP be _ that S	176
	NP be _ to VP[+EV]	268
	NP be _ to VP[-EV]	246

Table 1: # of verbs acceptable in each tense-frame pair based on our extension of MegaAcceptability.

replaced with *do*, *have*, or *happen*. This method aims to avoid unwanted typicality effects that might be introduced by interactions between our predicates of interest and more contentful items elsewhere in the sentence.[4]

We partition items into 248 lists of 32 items. Each list is constrained such that (i) 16 items had a first person subject, and 16 items had a third person subject; (ii) 16 items contain a *low frequency* verb and 16 items contain a *high frequency* verb, based on a median split of the frequencies in the SUBTLEX_US word frequency database (Brysbaert and New, 2009); (iii) 16 items are *low acceptability* and 16 items are *high acceptability*, based on a median split of the normalized acceptabilities for items selected from our extension of the MegaAcceptability dataset; (iv) no verb occurred more than once in the same list; (v) items containing a particular combination of matrix tense and syntactic frame occur in rough proportion to the number of verbs that are acceptable with that tense-frame combination based on our extension of the MegaAcceptability dataset (Table 1).

Participants 1,108 participants were recruited through Amazon Mechanical Turk to give 10 ratings per sentence in the 248 lists of 32—i.e. the end result contains 79,360 ratings for each of neg-raising and acceptability judgments. Participants were not allowed to respond to the same list more than once, though they were allowed to respond to as many lists as they liked. Each participants re-

[4]Because this method has not been previously validated for measuring neg-raising, we report two validation experiments in Appendix C, which demonstrate that the measure accords with judgments from prior work.

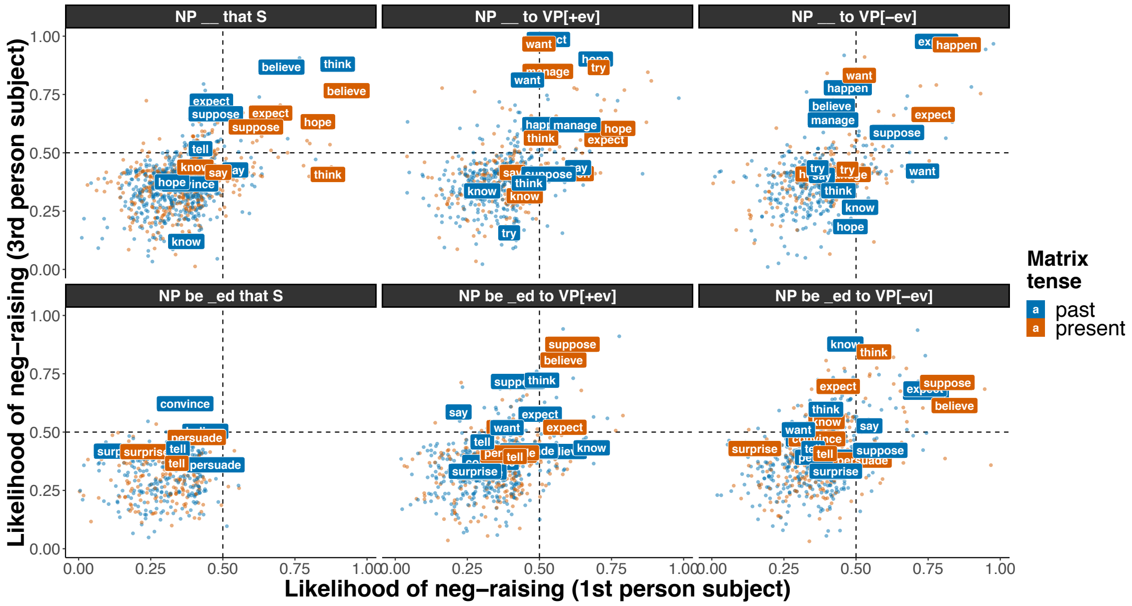


Figure 1: Normalized neg-raising scores for different subject, tense, and frame pairs.

sponded to 2.3 lists on average (min: 1, max: 16, median: 1). Of the 1,108 participants, 10 reported not speaking American English as their native language. Responses from these participants were filtered from the dataset prior to analysis. From this, responses for 27 lists were lost ($\sim$1% of the responses). This filtering removed at most two judgments for any particular item.

Results Figure 1 plots the normalized neg-raising scores for verbs in different subject (axes)-tense (color)-frame (block) contexts.[5] A verb (in some tense) being toward the top-right corner means that it shows strong neg-raising inferences with both first person and third person subjects, while a verb being towards the bottom-right corner means that it shows neg-raising behavior with first person subjects but not with third person subjects. The converse holds for the top-left corner: neg-raising behavior is seen with third person subjects but not first. We see that our method correctly captures canonical neg-raising predicates—e.g. *think* and *believe* with finite complements and *want* and *expect* with infinitival complements—as well as canonical non-neg-raising predicates—e.g. *know* and *say* with finite complements and *try* and *manage* with infinitivals.

4 Model

We aim to use our neg-raising dataset to assess which aspects of neg-raising inferences are due to properties of lexical items and which aspects are due to properties of the structures they compose with. To do this, we extend White and Rawlins' (2016) model of s(emantic)-selection, which induces semantic type signatures from syntactic distribution, with a module that associates semantic types with the *inference patterns* they trigger.

Our model has two hyperparameters that correspond to the theoretical constructs of interest: (i) the number of lexical properties relevant to neg-raising; and (ii) the number of structural properties relevant to neg-raising. In §5, we report on experiments aimed at finding the optimal setting of these two hyperparameters, and we analyze the parameters of the model fit corresponding to these hyperparameters in §6.

S-selection Model White and Rawlins' (2016) model of s-selection aims to induce verbs' semantic type signatures—e.g. that *love* can denote a relation between two entities and *think* can denote a relation between an entity and a proposition—from their syntactic distribution—e.g. that *love* is acceptable in NP _ NP frames and that *think* is acceptable in NP _ S frames. They formalize this task as a boolean matrix factorization (BMF) problem: given a boolean matrix $\mathbf{D} \in \mathbb{B}^{|\mathcal{V}|\times|\mathcal{F}|} = \{0,1\}^{|\mathcal{V}|\times|\mathcal{F}|}$, wherein $d_{vf} = 1$ iff verb $v \in \mathcal{V}$ is acceptable in syntactic frame $f \in \mathcal{F}$, one must induce boolean matrices $\mathbf{\Lambda} \in \mathbb{B}^{|\mathcal{V}|\times|\mathcal{T}|}$ and $\mathbf{\Pi} \in \mathbb{B}^{|\mathcal{T}|\times|\mathcal{F}|}$, wherein $\lambda_{vt} = 1$ iff verb v can have semantic type signature $t \in \mathcal{T}$ and $\pi_{tf} = 1$ iff t can be mapped onto syntactic frame f, such that (19): verb v is acceptable in frame f iff v has some type t that can be mapped (or *projected*) onto f.

[5] See Appendix D for details on normalization.

(19) $d_{vf} \approx \bigvee_t \lambda_{vt} \wedge \pi_{tf}$

As is standard in matrix factorization, the equivalence is approximate and is only guaranteed when there are as many semantic type signatures $\mathcal{T}$ as there are frames $\mathcal{F}$, in which case, the best solution is the one with $\mathbf{\Lambda} = \mathbf{D}$ and $\mathbf{\Pi}$ as the identity matrix of dimension $|\mathcal{T}| = |\mathcal{F}|$. Because this solution is trivial, $|\mathcal{T}|$ is generally much smaller than $|\mathcal{F}|$ and determined by fit to the data—in BMF, the count of how often $d_{vf} \neq \bigvee_t \lambda_{vt} \wedge \pi_{tf}$.

As an estimate of $\mathbf{D}$, White and Rawlins use the MegaAcceptability dataset, which we use in constructing our neg-raising dataset (§3). Instead of directly estimating the boolean matrices $\mathbf{\Lambda}$ and $\mathbf{\Pi}$, they estimate a probability distribution over the two under the strong independence assumption that all values λ_{vt} and π_{tf} are pairwise independent of all other values. This implies (20).[6]

(20) $\mathbb{P}(d_{vf}) = 1 - \prod_t 1 - \mathbb{P}(\lambda_{vt})\mathbb{P}(\pi_{tf})$

White and Rawlins treat $\mathbb{P}(d_{vf})$ as a fixed effect in an ordinal mixed effects model, which provides the loss function against which $\mathbb{P}(\lambda_{vt})$ and $\mathbb{P}(\pi_{tf})$ are optimized. They select the number of semantic type signatures to analyze by setting $|\mathcal{T}|$ such that an information criterion is optimized.

Neg-Raising Model We retain the main components of White and Rawlins' model but add a notion of *inference patterns* associated both with properties of verbs, on the one hand, and with semantic type signatures, on the other. In effect, this addition models inferences, such as neg-raising, as arising via a confluence of three factors: (i) properties of the relation a lexical item denotes—e.g. in a semantic/pragmatic approach, whatever property of a predicate triggers EM inferences; (ii) properties of the kinds of things that a predicate (or its denotation) relates—e.g. in a syntactic approach, whatever licenses "raising" of the negation; and (iii) whether a particular verb has a particular type signature. With respect to (ii) and (iii), it is important to note at the outset that, because we do not attempt to model acceptability, semantic type signatures play a somewhat different role in our model than in White and Rawlins': instead of determining which structures a verb is compatible with—i.e. (non)finite subordinate clauses, presence of a direct object, etc.—our model's type signatures control the inferences a particular verb can trigger when taking a particular structure. As such, our model's semantic type signatures might be more easily construed as properties of a structure that may or may not license neg-raising.[7] We thus refer to them as *structural properties*—in contrast to predicates' *lexical properties*.

Our extension requires the addition of three formal components to White and Rawlins' model: (i) a boolean matrix $\mathbf{\Psi} \in \mathbb{B}^{|\mathcal{V}|\times|\mathcal{I}|}$, wherein $\psi_{vi} = 1$ iff verb $v \in \mathcal{V}$ has property $i \in \mathcal{I}$; (ii) a boolean tensor $\mathbf{\Phi} \in \mathbb{B}^{|\mathcal{I}|\times|\mathcal{J}|\times|\mathcal{K}|}$, wherein $\phi_{ijk} = 1$ iff property i licenses a neg-raising inference with subject $j \in \mathcal{J}$ and tense $k \in \mathcal{K}$; and (iii) a boolean tensor $\mathbf{\Omega} \in \mathbb{B}^{|\mathcal{T}|\times|\mathcal{J}|\times|\mathcal{K}|}$, wherein $\omega_{tjk} = 1$ iff semantic type signature $t \in \mathcal{T}$ licenses a neg-raising inference with subject j and tense k.

As it stands, this formulation presupposes that there are both lexical and structural properties relevant to neg-raising. To capture the possibility that there may be only one or the other relevant to neg-raising, we additionally consider two families of *boundary models*. In the boundary models that posit no lexical properties—which (abusing notation) we refer to as $|\mathcal{I}| = 0$—we fix $\mathbf{\Psi} = \mathbf{1}_{|\mathcal{V}|}$ and $\mathbf{\Phi} = \mathbf{1}_{|\mathcal{I}|} \otimes \mathbf{1}_{|\mathcal{J}|} \otimes \mathbf{1}_{|\mathcal{K}|}$. In the boundary models that posit no structural properties ($|\mathcal{T}| = 0$) we fix $\mathbf{\Pi} = \mathbf{1}_{|\mathcal{F}|}$, $\mathbf{\Lambda} = \mathbf{1}_{|\mathcal{V}|}$, and $\mathbf{\Omega} = \mathbf{1}_{|\mathcal{T}|} \otimes \mathbf{1}_{|\mathcal{J}|} \otimes \mathbf{1}_{|\mathcal{K}|}$.

Analogous to White and Rawlins, we treat our task as a problem of finding $\mathbf{\Lambda}, \mathbf{\Pi}, \mathbf{\Psi}, \mathbf{\Phi}, \mathbf{\Omega}$ that best approximate the tensor $\mathbf{N}$, wherein $n_{vfjk} = 1$ iff verb v licenses neg-raising inferences in frame f with subject j and tense k. This is formalized in (21), which implies that $n_{vfjk} = 1$ iff there is some pairing of semantic type signature t and inference pattern i such that (i) verb v has semantic type signature t; (ii) verb v licenses inference pattern i; (iii) semantic type signature t can map onto frame f; and (iv) both t and i license a neg-raising inference with subject j and tense k.

(21) $n_{vfjk} \approx \bigvee_{t,i} \lambda_{vt} \wedge \psi_{vi} \wedge \phi_{ijk} \wedge \pi_{tf} \wedge \omega_{tjk}$

Also analogous to White and Rawlins, we aim to estimate $\mathbb{P}(n_{vfjk})$ (rather than n_{vfjk} directly) under similarly strong independence assumptions: $\mathbb{P}(\lambda_{vt}, \psi_{vi}, \phi_{ijk}, \pi_{tf}, \omega_{tjk}) = \mathbb{P}(\lambda_{vt})\mathbb{P}(\psi_{vi})$ $\mathbb{P}(\phi_{ijk})\mathbb{P}(\pi_{tf})\mathbb{P}(\omega_{tjk}) = \zeta_{vtifjk}$, implying (22).

(22) $\mathbb{P}(n_{vfjk}) = 1 - \prod_{t,i} 1 - \zeta_{vtifjk}$

We design the loss function against which $\mathbb{P}(\lambda_{vt})$,

[6] See Appendix E for the derivation of (20).

[7] Alternatively, they might be construed as (potentially cross-cutting) classes of syntactic structures and/or semantic type signatures that could be further refined by jointly modeling acceptability (e.g. as measured by MegaAcceptability) alongside our measure of neg-raising inferences.

$\mathbb{P}(\psi_{vi})$, $\mathbb{P}(\phi_{ijk})$, $\mathbb{P}(\pi_{tf})$, and $\mathbb{P}(\omega_{tjk})$ are optimized such that (i) $\mathbb{P}(n_{vfjk})$ is monotonically related to the neg-raising response r_{vfjkl} given by participant l for an item containing verb v in frame f with subject j and tense k (if one exists); but (ii) participants may have different ways of using the response scale. For example, some participants may prefer to use only values close to 0 or 1, while others may prefer values near 0.5; or some participants may be prefer lower likelihood values while others may prefer higher values. To implement this, we incorporate (i) a fixed scaling term σ_0; (ii) a fixed shifting term β_0; (iii) a random scaling term σ_l for each participant l; and (iv) a random shifting term β_l for each participant l. We define the expectation for a response r_{vfjkl} as in (23).

(23) $\hat{r}_{vfjkl} = \text{logit}^{-1}\left(m_a \nu_{vfjk} + \beta_0 + \beta_l\right)$
where $\nu_{vfjk} = \text{logit}\left(\mathbb{P}(n_{vfjk})\right)$
$m_l = \exp\left(\sigma_0 + \sigma_l\right)$

We optimize $\mathbb{P}(\lambda_{vt})$, $\mathbb{P}(\psi_{vi})$, $\mathbb{P}(\phi_{ijk})$, $\mathbb{P}(\pi_{tf})$, and $\mathbb{P}(\omega_{tjk})$ against a KL divergence loss, wherein r_{vfjkl} is taken to parameterize the true distribution and $\hat{r}_{vfjkl}$ the approximating distribution.

(24) $\mathrm{D}(r \parallel \hat{r}) = -\left[r \log \frac{\hat{r}}{r} + (1-r) \log \frac{1-\hat{r}}{1-r}\right]$

To take into account that it is harder to judge the neg-raising inferences for items that one cannot imagine hearing used, we additionally weight the above-mentioned KL loss by a normalization of the acceptability responses for an item containing verb v in frame f with subject j and tense k. We infer this value from the acceptability responses for an item containing verb v in frame f with subject j and tense k given by participant l, assuming a form for the expected value of a_{vfjkl} as in (25)—analogous to (23). (Unlike ν_{vfjk} in (23), α_{vfjk} in (25) is directly optimized.)

(25) $\hat{a}_{vfjkl} = \text{logit}^{-1}\left(m'_l \alpha_{vfjk} + \beta'_0 + \beta'_l\right)$
where $m'_l = \exp\left(\sigma'_0 + \sigma'_l\right)$

The final loss against which $\mathbb{P}(\lambda_{vt})$, $\mathbb{P}(\psi_{vi})$, $\mathbb{P}(\phi_{ijk})$, $\mathbb{P}(\pi_{tf})$, $\mathbb{P}(\omega_{tjk})$ are optimized is (26).[8]

(26) $\mathcal{L} = \sum \alpha'_{vfjk} \mathrm{D}(r_{vfjkl} \parallel \hat{r}_{vfjkl})$

where $\alpha'_{vfjk} = \text{logit}^{-1}(\alpha_{vfjk})$.

5 Experiment

We aim to find the optimal settings, relative to our neg-raising data, for (i) the number $|\mathcal{I}|$ of lexical properties relevant to neg-raising that it assumes; and (ii) the number $|\mathcal{T}|$ of structural properties relevant to neg-raising that it assumes. As with other models based on matrix factorization, higher values for $|\mathcal{I}|$ (with a fixed $|\mathcal{T}|$) or $|\mathcal{T}|$ (with a fixed $|\mathcal{I}|$) will necessarily fit the data as well or better than lower values, since a model with larger $|\mathcal{I}|$ or $|\mathcal{T}|$ can embed the model with a smaller value. However, this better fit comes at the cost of increased risk of overfitting due to the inclusion of superfluous dimensions. To mitigate the effects of overfitting, we conduct a five-fold cross-validation and select the model(s) with the best performance (in terms of our weighted loss) on held-out data.

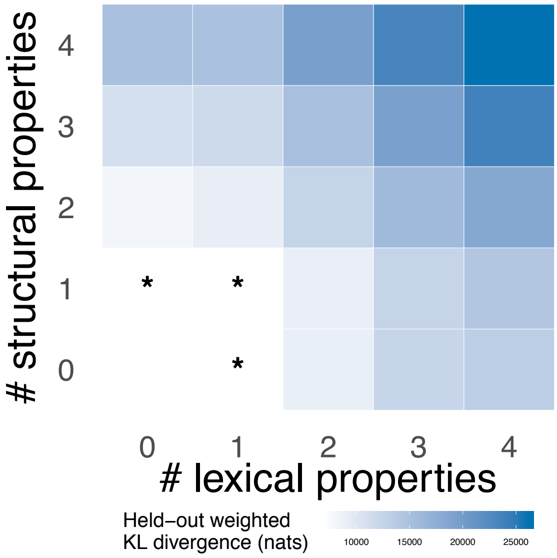


Figure 2: Sum of the weighted KL divergence loss across all five folds of the cross-validation for each setting of $|\mathcal{I}|$ (# of lexical properties) and $|\mathcal{T}|$ (# of structural properties). $|\mathcal{I}| = |\mathcal{T}| = 0$ was not run.

Method In this cross-validation, we pseudorandomly partition sentences from the neg-raising experiments into five sets (folds), fit the model with some setting of $|\mathcal{I}|, |\mathcal{T}|$ to the neg-raising responses for sentences in four of these sets (80% of the data), then compute the loss on the held-out set—repeating with each partition acting as the held-out set once. The assignment of items to folds is pseudorandom in that each fold is constrained to contain at least one instance of a particular verb with a particular complement type in some tense with some subject. If such a constraint were not enforced, on some folds, the model would have no data upon which to predict that verb with that complement. We consider each possible pairing of $|\mathcal{I}|, |\mathcal{T}| \in \{0, 1, 2, 3, 4\}$, except $|\mathcal{I}| = |\mathcal{T}| = 0$. The same partitioning is used for

[8] An additional term (not shown) is added to encode the standard assumption that the random effects terms are normally distributed with mean 0 and unknown variance.

every setting of $|\mathcal{I}|$ and $|\mathcal{T}|$, enabling paired comparison by sentence.

Implementation We implement our model in `tensorflow 1.14.0` (Abadi et al., 2016). We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.01 and default hyperparameters otherwise.

Results Figure 2 plots the sum of the weighted KL divergence loss across all five folds of the cross-validation for each setting of $|\mathcal{I}|$ (number of lexical properties) and $|\mathcal{T}|$ (number of structural properties). The best-performing models in terms of held-out loss (starred in Figure 2) are (in order): (i) one that posits one lexical property and no structural properties; (ii) one that posits no lexical properties and one structural property; and (iii) one that posits one lexical property and one structural property. None of these models' performance is reliably different from the others—as determined by a nonparametric bootstrap computing the 95% confidence interval for the pairwise difference in held-out loss between each pairing among the three—but all three perform reliably better than all other models tested.

Among these three, the model with the best fit to the dataset has $|\mathcal{I}| = 1$ and $|\mathcal{T}| = 1$. This result suggests that neg-raising is not purely a product of lexical knowledge: properties of the subordinate clause that a predicate combines with also influence whether neg-raising inferences are triggered. This is a surprising finding from the perspective of prior work, since (to our knowledge) no existing proposals posit that syntactic properties like the ones we manipulated to build our dataset—i.e. the presence or absence of tense, the presence or absence of an overt subject of the subordinate clause, and eventivity/stativity of a predicate in the subordinate clause—can influence whether neg-raising inferences are triggered. We next turn to analysis of this model fit to understand how our model captures patterns in the data.

6 Analysis

Table 2 gives the $|\mathcal{I}| = |\mathcal{T}| = 1$ model's estimate of the relationship between neg-raising inferences and lexical $\mathbb{P}(\phi_{ijk})$ (top) and structural properties $\mathbb{P}(\omega_{tjk})$ (bottom) with different subjects and tenses. The fact that all of the values in Table 2 are near 1 suggests that predicates having the lexical property or structures having the structural property will give rise to neg-raising inferences regardless of the subject and tense.[9]

		Tense	
Property	**Person**	*past*	*present*
lexical	*first*	0.93	0.98
	third	0.95	0.98
structural	*first*	0.93	0.98
	third	0.95	0.98

Table 2: Relationship between neg-raising inferences and lexical property $\mathbb{P}(\phi_{ijk})$ (top) and structural property $\mathbb{P}(\omega_{tjk})$ (bottom) with different subjects and tenses in $|\mathcal{I}| = |\mathcal{T}| = 1$ model.

This pattern is interesting because it suggests that the model does not capture the variability across different subjects and tenses observed in Figure 1 as a matter of either lexical or structural properties. That is, the model treats any variability in neg-raising inferences across different subjects and/or tenses as an idiosyncratic fact about the lexical item and the structure it occurs with—i.e. noise. This result makes intuitive sense insofar as such variability arises due to pragmatic reasoning that is specific to particular predicates, as opposed to some general semantic property.

But while the model does not distinguish among neg-raising inference with various subject and tense combinations, it does capture the coarser neg-raising v. non-neg-raising distinction among predicates—namely, by varying the probability that different lexical items have the lexical property $\mathbb{P}(\psi_{vi})$ and the probability that they select the structural property $\mathbb{P}(\lambda_{vt})$. Figure 3 plots the distribution of $\mathbb{P}(\psi_{vi}) \times \mathbb{P}(\lambda_{vt})$ across predicates.[10] We see that predicates standardly described as neg-raising (*think*, *believe*, *want*, *seem*, *feel*, etc.) fall to the right, while those standardly

[9]These tables appear to be copies of each other, but they are not. What is happening here is that the model is learning to associate $\mathbb{P}(\phi_{ijk})$ and $\mathbb{P}(\omega_{tjk})$ with (roughly) the square root of the largest expected value across all predicates for the neg-raising response to sentences with subject j and tense k. (It sets these values to the square root of the largest expected value because they will be multiplied together.) This strategy allows the model to simply vary $\mathbb{P}(\lambda_{vt})$, $\mathbb{P}(\psi_{vi})$, and $\mathbb{P}(\pi_{tf})$ to capture the likelihood a particular predicate or structure gives rise to neg-raising inferences, as described below.

[10]We plot the distribution of $\mathbb{P}(\psi_{vi}) \times \mathbb{P}(\lambda_{vt})$, instead of showing a scatter plot, because these probabilities show extremely high positive rank correlation—approximately 1. This happens because, when there is only one lexical property and one structural property, the lexical property and selection probabilities are effectively a single parameter p, with $\mathbb{P}(\psi_{vi})$ and $\mathbb{P}(\lambda_{vt})$ themselves being set to $\sqrt{p}$ (see also Footnote 9).

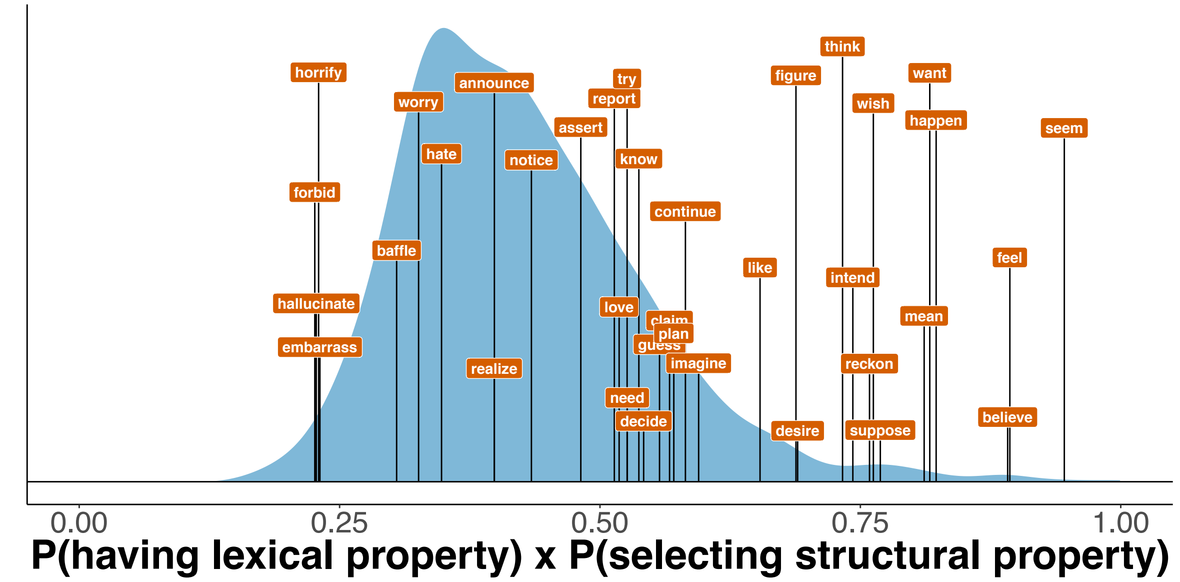


Figure 3: Distribution of $\mathbb{P}(\psi_{vi}) \times \mathbb{P}(\lambda_{vt})$ across predicates, along with selected neg-raising (toward right) and non-neg-raising (toward left) predicates in $|\mathcal{I}| = |\mathcal{T}| = 1$ model. (Label height is jittered to avoid overplotting.)

described as non-neg-raising (*know*, *notice*, *realize*, *love*, etc.) fall to left. Thus, in some sense, a predicate's probability of having the model's single lexical property (plus its probability of selecting the single structural property) appears to capture something like the probability of neg-raising.

Structure	**Probability**
NP __ that S	0.91
NP be _ed that S	0.84
NP __ to VP[+ev]	0.98
NP be _ed to VP[+ev]	0.93
NP __ to VP[-ev]	0.94
NP be _ed to VP[-ev]	0.98

Table 3: Relationship between structural property and structures $\mathbb{P}(\pi_{tf})$ in $|\mathcal{I}| = |\mathcal{T}| = 1$ model.

The model captures variability with respect to different syntactic structures by modulating $\mathbb{P}(\pi_{tf})$, shown in Table 3. Looking back to Figure 1, these values roughly correlate with the largest neg-raising response (across subjects and tenses) seen in that frame, with NP be _ed that S showing the lowest such value. The value of $\mathbb{P}(\pi_{tf})$ is not the *same* as the largest neg-raising value in Figure 1, likely due to the fact that many of the predicates that occur in that frame also have small values for $\mathbb{P}(\psi_{vi}) \times \mathbb{P}(\lambda_{vt})$, and thus, when $\mathbb{P}(\pi_{tf})$ is multiplied by that values, it is small.

7 Conclusion

We presented a probabilistic model to induce the mappings from lexical sources and their grammatical sources to neg-raising inferences. We trained this model on a large-scale dataset of neg-raising judgments that we collected for 925 English clause-embedding verbs in six distinct syntactic frames as well as various matrix tenses and subjects. Our model fit the best when positing one lexical property and one structural property. This is a surprising finding from the perspective of prior work, since (to our knowledge) no existing proposals posit that syntactic properties like the ones we manipulated to build our dataset—i.e. the presence or absence of tense, the presence or absence of an overt subject of the subordinate clause, and eventivity/stativity of a predicate in the subordinate clause—can influence whether neg-raising inferences are triggered. Our findings suggest new directions for theoretical research attempting to explain the interaction between lexical and structural factors in neg-raising. Future work in this vein might extend the model proposed here to investigate the relationship between neg-raising and acceptability as well as other related phenomena with associated large-scale datasets, such as lexically triggered veridicality inferences (White and Rawlins, 2018; White et al., 2018; White, 2019).

Acknowledgments

We would like to thank the FACTS.lab at UR as well as three anonymous reviewers for useful comments. This work was supported by an NSF grant (BCS-1748969/BCS-1749025) *The MegaAttitude Project: Investigating selection and polysemy at the scale of the lexicon.*

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Dorit Abusch. 2002. Lexical alternatives as a source of pragmatic presuppositions. *Semantics and Linguistic Theory*, 12:1–19.

Renate Bartsch. 1973. "Negative transportation" gibt es nicht. *Linguistische Berichte*, 27(7).

Marc Brysbaert and Boris New. 2009. Moving beyond kucera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41:977–990.

Gennaro Chierchia. 2013. *Logic in grammar: polarity, free choice, and intervention*, first edition. Oxford University Press, Oxford.

Chris Collins and Paul Martin Postal. 2014. *Classical NEG Raising: An Essay on the Syntax of Negation*. MIT Press.

Chris Collins and Paul Martin Postal. 2017. Interclausal neg raising and the scope of negation. *Glossa: A Journal of General Linguistics*, 2:1–29.

Chris Collins and Paul Martin Postal. 2018. Disentangling two distinct notions of neg raising. *Semantics and Pragmatics*, 11(5).

Paul Crowley. 2019. Neg-raising and neg movement. *Natural Language Semantics*, 27(1):1–17.

Mark Davies. 2017. Corpus of Contemporary American English (COCA).

Charles J. Fillmore. 1963. The position of embedding transformations in a grammar. *WORD*, 19(2):208–231.

Jon R. Gajewski. 2007. Neg-raising and polarity. *Linguistics and Philosophy*, 30(3):289–328.

Jon R. Gajewski. 2011. Licensing strong npis. *Natural Language Semantics*, 19(2):109–148.

Vincent Homer. 2015. Neg-raising and positive polarity: The view from modals. *Semantics and Pragmatics*, 8(4):1–88.

Laurence Robert Horn. 1971. Negative transportation: unsafe at any speed? *Papers from the seventh regional meeting, Chicago Linguistic Society*, pages 120–133.

Laurence Robert Horn. 1978. Remarks on neg-raising. *Syntax and Semantics*, 9:129–220.

Laurence Robert Horn. 1989. *A Natural History of Negation*. University of Chicago Press.

Laurence Robert Horn. 2014. The cloud of unknowing. In Jack Hoeksema and Dicky Gilbers, editors, *Black Book: A Festschrift for Frans Zwarts*, pages 178–196. University of Groningen, Groningen, the Netherlands.

Laurence Robert Horn and Samuel Bayer. 1984. Short-circuited implicature: A negative contribution. *Linguistics and Philosophy*, 7(4):397–414.

Ray S. Jackendoff. 1971. On some questionable arguments about quantifiers and negation. *Language*, 47(2):282–297.

Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, and Vít Suchomel. 2014. The sketch engine: ten years on. *Lexicography*, pages 7–36.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*.

Paul Kiparsky. 1970. Semantic rules in grammar. In Hreinn Benediktsson, editor, *The Nordic Languages and Modern Linguistics*, pages 262–285. Visindafelag Islendinga, Reykjavik.

Edward S. Klima. 1964. *Negation in English*. Englewood Cliffs, NJ: Prentice-Hall.

Howard Lasnik. 1972. *Analyses of Negation in English*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Jay M. Pollack. 1976. A re-analysis of neg-raising in english. *Working Papers in Linguistics*, 21:189–239.

Jacopo Romoli. 2013. A scalar implicature-based approach to neg-raising. *Linguistics and Philosophy*, 36(4):291–353.

Jacopo Romoli and Matthew Mandelkern. 2019. What's not to like. *Linguistic Inquiry*, 0(ja):1–21.

Nadine Theiler, Floris Roelofsen, and Maria Aloni. 2017. What's wrong with believing whether? *Semantics and Linguistic Theory*, 27:248–265.

Lucia M. Tovena. 2001. Neg-raising: negation as failure. In Jack Hoeksema, Hotze Rullmann, Víctor Sánchez-Valencia, and Ton van der Wouden, editors, *Perspectives on Negation and Polarity Items*, pages 331–356. John Benjamins, Amsterdam.

Aaron Steven White. 2019. Lexically triggered veridicality inferences. To appear in *Handbook of Pragmatics*.

Aaron Steven White and Kyle Rawlins. 2016. A computational model of s-selection. *Semantics and Linguistic Theory*, 26:641–663.

Aaron Steven White and Kyle Rawlins. 2018. The role of veridicality and factivity in clause selection. *Proceedings of the 48th Meeting of the North East Linguistic Society*.

Aaron Steven White and Kyle Rawlins. 2019. Frequency, acceptability, and selection: A case study of clause-embedding. Under revision for *Glossa*.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2018. Lexicosyntactic inference in neural models. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724.

Yimei Xiang. 2013. Neg-raising: Focus and implicatures. *Proceedings of Sinn und Bedeutung*, 18:487–503.

Hedde Zeijlstra. 2018. Does neg-raising involve neg-raising? *Topoi*, 37(3):417–433.

Frans Zwarts. 1998. Three types of polarity. In Fritz Hamm and Erhard Hinrichs, editors, *Plurality and Quantification*, pages 177–238. Springer Netherland, Dordrecht.

A Instructions

In this experiment, you will be asked to answer questions about what a person is likely to mean if they say a particular sentence.

Your task will be to respond about the likelihood on the slider that will appear under each question, where the left side corresponds to *extremely unlikely* and the right side corresponds to *extremely likely*.

For instance, you might get the question *If I were to say John has three kids, how likely is it that I mean John has exactly three kids?* with a slider. In this case you would move the slider handle fairly far to the right (toward *extremely likely*), since if someone says "John has three kids", it's pretty likely that they mean that John has exactly three children.

If the question were *If I were to say some of the boys left, how likely is it that I mean all of the boys left?*, then you might move the slider pretty far to the left (toward *extremely unlikely*), since it would be odd if someone says "Some of the boys left – and by that, I mean all of the boys left".

And if the question were *If I were to say Ann didn't greet everyone politely, how likely is it that I mean Ann was unwelcoming to every single person?*, you might leave the slider in the middle (which corresponds to *maybe or maybe not*), since quite often such sentence can be used to mean Ann greeted some people politely but not all, or to mean Ann was not polite to every single person.

Try to answer the questions as quickly and accurately as possible. Many of the sentences may not be sentences that you can imagine someone ever saying. Try your best to interpret what a speaker would mean in using them. After each question, you will be given a chance to tell us whether the sentence you just responded to isn't something you can imagine a native English speaker ever saying.

Not all questions have correct answers, but a subset in each HIT do. Prior to approval, we check the answers given for this subset. We will reject work containing a substantial number of answers that do not agree with the correct answer.

When the experiment is over, a screen will appear telling you that you are done, and a submission button will be revealed.

B Data

We extend White and Rawlins' (2016) MegaAcceptability v1.0 dataset by collecting acceptability judgments for sentences in present and past progressive tenses—resulting in MegAcceptability v2.0, which subsumes MegaAcceptability v1.0. To enable comparison of the judgments given in MegaAcceptability v1.0 and those we collect, we run an additional *linking experiment* with half items from MegaAcceptability v1.0 and our extension. We then normalize all three datasets separately using the procedure described in White and Rawlins 2019 and then combine them by using the linking experiment data to train a model to map them into a comparable normalized rating space. Both the extended MegaAcceptability and linking datasets are available at megaattitude.io.

Extended MegaAcceptability Our test items are selected and modified from the top 25% most acceptable verb-frame pairs from the MegaAcceptability dataset of White and Rawlins (2016), determined by a modified version of the normalization procedure used in White and Rawlins 2019. This item set thus contains 12,500 verb-frame pairs, with 1000 unique verbs and the same 50 subcategorization frames (35 in active voice and 15 in passive voice) that are used in MegaAcceptability.

Given the 12,500 verb-frame pairs, we construct new sentences in both present and past progressive tense/aspect, resulting in a total of 25,000 items. Examples of two sentences from MegaAcceptability v1.0 are given in (27) and the corresponding present and past progressive versions are given in (28) and (29), respectively.

(27) a. Someone knew which thing to do.
b. Someone talked about something.

(28) a. Someone knows which thing to do.
b. Someone talks about something.

(29) a. Someone is knowing which thing to do.
b. Someone was talking about something.

All methods follow White and Rawlins 2016. Sentences are partitioned into 500 lists of 50, with each list constructed such that (i) each frame shows up once in a list, making each list contain 50 unique frames, if possible; (ii) otherwise, the distribution of frames are kept as similar as possible across lists; and (iii) no verbs appear more than once in a list. We gather 5 acceptability judgments per sentence, yielding a total of 125,000 judgments for 25,000 items.

Judgments for each sentence in a list are collected on a 1-to-7 scale. To avoid typicality effects, we construct the frames to contain as little lexical content as possible besides the verb at hand. For this, we instantiate all NP arguments as indefinite pronouns *someone* or *something* and all verbs besides the one being tested as *do* or *happen*. 565 participants were recruited from Amazon Mechanical Turk, where 562 speak American English as their native language.

Linking experiment Because our extension of MegaAcceptability was built in such a way that it likely contains higher acceptability items, the ratings in MegaAcceptability v1.0 and the ratings in our extension are likely not comparable—i.e. a rating in MegaAcceptability v1.0 is, in some sense, a worse rating than in our extension, since our sentences are, by construction, better overall. To put the existing MegaAcceptability dataset and our extended dataset on a comparable scale, we run another experiment to assist in mapping the two datasets to such a comparable scale. We choose 50 items, each with a unique verb, by selecting 26 items from our dataset (14 in present tense and 12 in past progressive tense) and 24 items from MegaAcceptability (all past tense).

This item selection was constrained such that half of the items chosen were below the median acceptability score and half were above, evenly split across items from our experiment and items from MegaAcceptability v1.0. The items with the lowest acceptability scores consist of 8 in the present, 6 in the past progressive, and 12 in the past tense and so do the items with the highest acceptability scores. Example items with the low acceptability scores (under this criterion) are shown in (30), and example items with high acceptability scores are shown in (31).

(30) a. Someone demands about whether something happened.
b. Someone was judging to someone that something happened.
c. Someone invited which thing to do.

(31) a. Someone is distracted.
b. Someone was teaching.
c. Someone dared to do something.

The linking experiment is built in a very similar manner to our extension of MegaAcceptability, described above. Ordinal acceptability judgments are collected on a 1-to-7 scale. 50 participants were recruited to rate all 50 items in the experiment. All of the 50 participants report speaking American English as their native language.

After running the linking experiment, we normalize the ratings in all three datasets separately using a modified version of the procedure described in White and Rawlins 2019. Then, we construct one mapping from the normalized ratings in our extension of MegaAcceptability to the normalized ratings for the linking dataset and another mapping from the normalized ratings in the linking dataset to the normalized ratings in MegaAcceptability v1.0 with two linear regressions—implemented in `scikit-learn` (Pedregosa et al., 2011). We then compose these two regressions to map the normalized ratings in our extended MegaAcceptability dataset to those in MegaAcceptability v1.0. This gives us a combined dataset of acceptability judgments for sentences in three different tense/aspect combinations

Subordinate clause	Neg-raising	Non-neg-raising
Finite	think, believe, feel, reckon, figure, guess, suppose, imagine	announce, claim, assert, report, know, realize, notice, find out
Infinitival	want, wish, happen, seem, plan, intend, mean, turn out	love, hate, need, continue, try, like, desire, decide

Table 4: Verbs used in validation experiments

(*past, present*, and *past progressive*) and 50 different syntactic frames, which we use to construct our neg-raising experiment.

C Validation Experiments

We conduct experiments aimed at validating our method for measuring neg-raising. In both experiments, we test the same set of 32 clause-embedding verbs, half of which we expect to show neg-raising behavior and the other half we do not (based on the literature discussed in §2). For neg-raising verbs, we refer to the neg-raising predicates listed in Gajewski 2007 and Collins and Postal 2018; and for non-neg-raising verbs, we choose factive verbs and those that Theiler et al. (2017) claim are not neg-raising. The experiments differ with respect to whether we employ "bleached" items (as in the data collection described in the main body of the paper) or "contentful" items, which are constructed based on sentences drawn from English corpora.

Materials We select neg-raising and non-neg-raising verbs such that half of each type takes infinitival subordinate clauses and half takes finite subordinate clauses. Table 4 shows the 32 verbs we choose for the pilot. Some verbs listed as taking one kind of subordinate clause can also take the other. In these cases, we only test that verb in the subordinate clause listed in Table 4.

The matrix subject (first v. third person) and matrix tense (present v. past) are manipulated for each predicate: (32) schematizes four items from our bleached experiment and (33) schematizes four items from our contentful experiment.

(32) {I, A particular person} {don't/doesn't, didn't} want to do a particular thing.

(33) {I, Stephen} {don't/doesn't, didn't} want to introduce new rules.

Items for the bleached experiment are constructed automatically using the templates, which select *to have a particular thing* for *turn out* and *seem* as their subordinate clause, *to do a particular thing* for other verbs taking infinitival subordinate clauses, and *that something happened* for the verbs taking finite subordiante clauses. Items for the contentful experiment are constructed by replacing all bleached words (*a particular person, a particular thing, do, have*, and *happen*) from the bleached experiment items by contentful lexical words.

The high content sentences are constructed based on sentences sampled from the Corpus of Contemporary American English (Davies, 2017) and the Oxford English Corpus (Kilgarriff et al., 2014). The contentful items are modified so that third person subject is a proper name and sentences do not include any pauses or conjunctions. To allow possible item variability, we create five contentful items per each bleached item.

For the bleached experiment, four lists of 32 items each are constructed by partitioning the resulting 128 items under the constraints that (i) every list contains every verb with exactly one subject (*first*, *third*) and tense (*past*, *present*) and (ii) every subject-tense pair is seen an equal number of times across verbs. We ensure that the same level of a particular factor is never assigned to the same verb more than once in any list and that the items in a list are randomly shuffled. To construct items, we manipulate neg-raising, embedded complement, matrix subject, matrix tense. Neg-raising and embedded complements are pre-determined for each verb, while matrix subject and matrix tense are randomly selected for a verb in each task. The same constraints apply for the contentful experiment except that the test items were partitioned into 20 lists of 32 instead of four lists because the total number of sentences for the contentful experiment is five times bigger than the bleached experiment.

Participants For the bleached experiment, 100 participants were recruited such that each of the four lists was rated by 25 unique participants. For the contentful experiment, 100 participants were recruited as well, to have each of the 20 lists of 32 rated by five unique participants. No participant was allowed to rate more than one list. In each experiment, one participant out of 100 reported not speaking American English natively and this participant's responses were filtered prior to analysis.

Analysis We test whether our task correctly captures canonical (non-)neg-raising verbs using linear mixed effects models. For both validation experiments, we start with a model containing fixed effects for NEGRAISING (*true*, *false*; as in Table 4), random intercepts for PARTICIPANT, VERB, and (in the contentful validation) ITEM. Nested under both verb and participant, we also included random intercepts for MATRIX SUBJECT (*1st*, *3rd*) and MATRIX TENSE (*past*, *present*) and their interaction. We compare this against a model with the same random effects structure but no effect of NEGRAISING. We find a reliably positive effect of NEGRAISING for both the bleached experiment ($\chi^2(1) = 34.5$, $p < 10^{-3}$) and the contentful experiment ($\chi^2(1) = 19.8$, $p < 10^{-3}$). This suggests that participants' responses are consistent with neg-raising inferences being more likely with verbs that have previously been claimed to give rise to such inferences.

D Normalization

For the purposes of visualization in §3, we present normalized neg-raising scores. These scores are derived using a mixed effects robust regression with loss the same loss (26) as for the model described in Section 4, except that, unlike for the model, where ν_{vfjk} is defined in terms of the model, for the purposes of normalization, both ν_{vfjk} in (23) and α_{vfjk} in (25) are directly optimized. Figure 1 plots $\text{logit}^{-1}(\exp(\sigma_0)\nu_{vfjk})+\beta_0$.

E Model Derivation

$$\begin{aligned}
\mathbb{P}(d_{vf}) &= \mathbb{P}\left(\bigvee_t \lambda_{vt} \wedge \pi_{tf}\right) \\
&= \mathbb{P}\left(\neg\neg\bigvee_t \lambda_{vt} \wedge \pi_{tf}\right) \\
&= \mathbb{P}\left(\neg\bigwedge_t \neg(\lambda_{vt} \wedge \pi_{tf})\right) \\
&= \mathbb{P}\left(\neg\bigwedge_t \neg(\lambda_{vt} \wedge \pi_{tf})\right) \\
&= 1 - \mathbb{P}\left(\bigwedge_t \neg(\lambda_{vt} \wedge \pi_{tf})\right) \\
&= 1 - \prod_t \mathbb{P}\left(\neg(\lambda_{vt} \wedge \pi_{tf})\right) \\
&= 1 - \prod_t 1 - \mathbb{P}\left(\lambda_{vt} \wedge \pi_{tf}\right) \\
&= 1 - \prod_t 1 - \mathbb{P}(\lambda_{vt})\mathbb{P}(\pi_{tf})
\end{aligned}$$

Communication-based Evaluation for Natural Language Generation

Benjamin Newman Stanford University
Reuben Cohn-Gordon Stanford University
Christopher Potts Stanford University
{ blnewman, reubencg, cgpotts } @stanford.edu

Abstract

Natural language generation (NLG) systems are commonly evaluated using n-gram overlap measures (e.g. BLEU, ROUGE). These measures do not directly capture semantics or speaker intentions, and so they often turn out to be misaligned with our true goals for NLG. In this work, we argue instead for *communication-based* evaluations: assuming the purpose of an NLG system is to convey information to a reader/listener, we can directly evaluate its effectiveness at this task using the Rational Speech Acts model of pragmatic language use. We illustrate with a color reference dataset that contains descriptions in predefined quality categories, showing that our method better aligns with these quality categories than do any of the prominent n-gram overlap methods.

1 Introduction

Natural language generation (NLG) models are increasingly prominent as core components in dialogue agents, story generators, summarization tools, image captioning systems, and others. NLG models are generally evaluated according to metrics that are defined in terms of the n-gram overlap between the model-generated candidate and human-generated reference texts. However, these metrics suffer from a well-known limitation: they assume that quality candidates will always share many exact token matches with ones generated by humans. This assumption is false for many common linguistic phenomena. For example, synonymous expressions receive low scores with most of these metrics even though humans find them equally good, and negated candidates receive high scores even where the negation leads to dramatic deviation from the reference texts. Such metrics are particularly ineffective in scenarios where there are many potentially appropriate utterances (Liu et al., 2016; Novikova et al., 2017).

To avoid this problem, one might turn to human judgments to assess the quality of model-generated language. In this setting, humans rate language according to grammaticality, typicality, informativeness, interestingness, and other qualitative dimensions (Lowe et al., 2017; Hashimoto et al., 2019; Chaganty et al., 2018). This addresses the problems with n-gram overlap methods, but it is expensive, and the human task does not reflect natural language use, which can lead to unreliable data.

One shortcoming of these methods is that they fail to take into account the communicative function of language; a speaker's goal is not only to produce well-formed expressions, but also to convey relevant information to a listener. Likewise, a listener is not only an assessor of quality, but also an agent that forms beliefs based on speakers' utterances. Thus, our NLG systems should be expected to use language to communicate as well, and we should evaluate these systems, not based on surface-level features of their utterances, but rather on the information they convey.

In this work, we argue for such *communication-based* evaluations. In language use, the speaker intends to communicate information to the listener using an utterance, and the listener infers some information from that utterance. This provides the basis for evaluation: if the listener's inference aligns with the speaker's intentions, the utterance was successful. If these intentions are not aligned, the utterance was less successful.

We formalize communication-based NLG evaluations using the Rational Speech Acts model of pragmatic language use (Frank and Goodman, 2012). To motivate this approach, we rely on a color reference game (Monroe et al., 2017, 2018). In this game, a speaker and a listener see a set of three colors. The speaker is told one color is the target and tries to communicate the target to the listener using a natural language utterance. A good utterance is more likely to lead the listener to select the target, while a bad utterance is less likely to do so. In turn, effective metrics should assign high scores to good utterances and low scores to bad ones.

To test our evaluation proposal, we asked crowdworkers to write color descriptions falling into three separate quality categories: those that describe only the target color (*descriptive* candidates); those that describe the target color and at least one other color in the context (*ambiguous* candidates); and those that describe only one non-target color in the context (*misleading* candidates). We then assess the extent to which our method's scores align with these categories. For comparison, we also investigate the extent to which n-gram overlap metrics correlate with utterance quality, focusing specifically on BLEU, METEOR, ROUGE, and CIDEr. We find that our communication-based metrics correlate more strongly than n-gram overlap metrics do. Our findings suggest that, when evaluating NLG models grounded in a task, it is more effective to use task performance than n-gram overlap metrics.

2 Related Work

2.1 NLG Evaluation

Existing NLG evaluation methods make use of n-gram overlap scores, human evaluations, and model-based evaluations. Our own method blends human evaluation and model-based evaluation, as we advocate using humans or building models to act on generated language

Other model-based evaluations take a variety of forms. Some involve training models to estimate human judgments of utterance quality (Lowe et al., 2017; Dušek et al., 2017; Kann et al., 2018). Others require training models to distinguish between language generated by humans and models—an adversarial evaluation (Bowman et al., 2016; Liu et al., 2016; Kannan and Vinyals, 2016; Bruni and Fernández, 2017). These methods focus on the utterance in a vacuum and tend to not to consider how language will actually interact with other conversational participants. They treat humans as assessors of quality or adversarial listeners, whereas our proposal takes the perspective that listeners are cooperative interlocutors who use the language they hear to inform their beliefs about the world.

Our approach can also be seen as part of a larger effort to incorporate context into NLG evaluation. Prior work in this area includes the image captioning metric SPICE, which uses scene graphs to assess candidate captions (Anderson et al., 2016). Similarly, Lowe et al. (2017) use conversational context to predict how human annotators would score dialogue agents, and the importance of context in assessment of this domain is noted by Liu et al. (2016). Our work incorporates contextual information by modeling the task a hypothetical listener will perform with the language produced.

2.2 Task-based Language Evaluation

Our work is particularly relevant for evaluation of utterances in task-specific scenarios. Overwhelmingly, work in this area uses humans performing some task with model-generated utterances to evaluate these utterances (Andreas and Klein, 2016; Andreas et al., 2017; Golland et al., 2010; Mao et al., 2016; Vedantam et al., 2017). Additionally, automatic evaluation metrics have been proposed. Monroe et al. (2017) and Cohn-Gordon et al. (2018) use a combination of language models conditioned on the context and Bayes' rule, while Mao et al. (2016) use their joint image and text classifier to evaluate potential object descriptions. We compare these two approaches as well. Additionally, referring expressions tend not to be evaluated using n-gram overlap metrics; Vedantam et al. (2017)'s use of CIDEr is an exception. As far as we know, these communication-based and n-gram overlap evaluation approaches have not previously been compared.

2.3 Communicative Informativity

Our communication-based evaluation method is closely related to the Rational Speech Acts (RSA) framework of pragmatic language use. This framework describes communication between two agents as a rational act where one agent, the speaker, chooses to communicate some information to another agent, the listener. The speaker chooses their utterance to maximize their utility, which in the framework involves choosing the utterance most helpful to the listener (Goodman and Frank, 2016). This idea has been used to model a wide range of linguistic phenomena.

This utility function is very similar to our proposed method's scoring function—differing only in a cost term. To our knowledge, this is the first case where this rational speaker utility function is used to evaluate language rather than model human utterance selection.

3 N-gram Overlap Evaluation Metrics

We now introduce the n-gram overlap metrics we adopt as baselines for our evaluations. These metrics evaluate candidate utterances by identifying the n-grams shared between the candidate utterances and human-generated reference utterances. They are commonly used for evaluation in a variety of domains and are consistently compared when evaluating the effectiveness of different metrics for various tasks (summarization, image captioning, dialogue; Novikova et al. 2017; Kilickaya et al. 2017; Sharma et al. 2017).

BLEU BLEU (BiLingual Evaluation Understudy) was conceived as a method for automatically evaluating machine translation systems by comparing the tokens in the system outputs to reference sentences constructed by expert translators (Papineni et al., 2002). BLEU consists of two components—a modified n-gram precision and a brevity penalty. The modified n-gram precision rewards candidate translations that contain the same n-grams as the references. Calculated precisions for n-grams of different sizes are then geometrically averaged together. Conventionally, n-gram overlaps for n = 1, 2, 3, and 4 are calculated. The second component of the BLEU score, the brevity penalty, acts as a recall constraint. Long candidate utterances could achieve a high modified n-gram precision by containing many n-grams, but the brevity penalty negatively impacts the score of candidates longer than the reference.

METEOR METEOR (Metric for Evaluation of Translation with Explicit ORdering), like BLEU, is designed for assessing utterances generated by machine translation systems (Banerjee and Lavie, 2005). METEOR searches for an alignment between the candidate and reference sentence using a form of beam search. Stemmed words, synonyms, and even paraphrases are considered in seeking the optimal alignment. This alignment is used to a calculate an F-score, usually favoring recall over precision. METEOR also has a “fragmentation score” that penalizes non-contiguous alignments and addresses issues related to word order. High METEOR scores mean large overlap between the tokens in the reference and candidate (including synonymy) as well as the correct word order.

ROUGE ROUGE (Recall Oriented Understudy of Gisting Evaluation) is a class of n-gram overlap metrics for assessing summaries (Lin, 2004). Like BLEU, many ROUGE metrics operate on the n-gram level, but unlike BLEU, their main component is an n-gram recall score that gives the proportion of n-grams in a reference that are in the candidate rather than a precision score that gives the proportion of n-grams in the candidate that are in the reference. The version of ROUGE we use here is called ROUGE-L. It uses the longest common subsequence between the candidate summary and reference summary to calculate an F-score heavily favoring recall. As such, a high ROUGE-L score indicates that a large proportion of tokens from the reference occur in the candidate, so longer candidates are rewarded (Vedantam et al., 2015).

CIDEr CIDEr (Consensus-based Image Description Evaluation) is an n-gram overlap metric that assesses image captions (Vedantam et al., 2015). It attempts to capture how well a candidate agrees with the “consensus” of a large group of references. It does this by creating TF-IDF vectors for different n-grams sizes from the reference and candidate captions and calculating a weighted average of the cosine similarities between vectors for different n-gram sizes. Inverse document frequency is calculated over all of the reference sentences in the dataset. High CIDEr scores indicate that a candidate caption uses the same infrequent, and likely informative, n-grams as a number of the references.

While the metrics described above (other than CIDEr) are defined for a single candidate and a single reference, the intention is that they be used with multiple reference texts per candidate, and Finch et al. (2004) found that using more reference sentences increases the reliability of these metrics. Gains start at 4 and continue up until 50 reference sentences in some cases (Vedantam et al., 2015). This is because a greater number of references provides more opportunities for the candidate to get a higher score. Because of this, when comparing these metrics to our communication-based evaluation we use multiple references.

4 Communication-based Evaluation

We now define our communication-based evaluation method in general terms, leaving its specific application to the color reference game to Section 5.

For our evaluation, we treat an NLG system as a speaker attempting to communicate about a topic t. We denote the set of all world states relevant to t as W_t, with a random world state drawn from this set represented as $w_t \in W_t$. These world states reflect any aspect of the world a speaker might want to communicate. While this set is potentially infinite, the topic t limits the set to just contain alternatives relevant to what commands the speaker's attention. The speaker's knowledge of the state of the world relevant to their communicative topic can then be represented as a distribution over world states, $S(w_t)$, as they may have different confidence levels about different alternatives. The speaker's goal is to communicate their distribution to the listener with an utterance u. After hearing u, the listener has some beliefs about the same topic-relevant states, which we can represent with the conditional distribution $L(w_t \mid u)$. This distribution signifies the listener's representation of the world related to t, so, if the speaker is successful, $L(w_t \mid u)$ should be close to $S(w_t)$. As such, we can define our method M to measure the similarity between these two distributions with the KL-divergence:

$$M(u \mid S, L) = D_{KL}(S(w_t) \parallel L(w_t \mid u)) \quad (1)$$

If the speaker has a specific target state w_{target} in mind, then all of the speaker's probability mass is on w_{target}. In that situation, the KL-Divergence is equivalent to the negative log-likelihood of the listener's probability of the true target color being the target:

$$M(u \mid L, S) = -\log L(w_{\text{target}} \mid u) \quad (2)$$

This value directly quantifies the listener's accuracy in guessing the speaker's target state.

This metric can also be seen as measuring the communicative informativeness of u in the sense of RSA. As described previously, in this framework, a speaker chooses an utterance to maximize their utility. Conventionally, this utility is the quantity represented by our metric—the KL-divergence between the speaker's observed distribution of the world and what they expect their listener's distribution to be after hearing the potential utterance. In this way, our approach can be seen as defining NLG quality in terms of pragmatic language use.

As is evident from this description, our method requires NLG systems to be construed as producing utterances that would help a listener distinguish among relevant alternative states. For example, an image captioning system has to be designed, not just to create true captions for its input images, but also to create captions that would help a listener choose that input from among a set of distractor images (Vedantam et al., 2017; Mao et al., 2016; Cohn-Gordon et al., 2018). Similarly, a summarization tool should produce summaries that capture exactly the information that makes the source text stand out with respect to related inputs (Zhang et al., 2018), and a pure text generation tool (a language model) might be refashioned to produce texts conditional on specific pieces of metadata (e.g. genre, author) so that we can assess it based on a listener's ability to recover that metadata from distractors (Shen et al., 2019). In general, we feel that these are healthy impositions on these tasks, as they encourage the systems to be grounded in specific contexts and to produce utterances that are not just true but also informative.

5 Evaluating the Evaluation Approaches

We assess the effectiveness of our evaluation method using the color reference game described by Monroe et al. (2017), in which a speaker and a listener each see the same set of three color swatches (though perhaps in different orders) and the speaker's task is to convey the identity of their (hidden) target color to the listener. This scenario is ideal for our evaluation because the communicative goal is clear, and we can easily adjust this goal in ways that affect utterance quality.

5.1 Data

Our assessment hinges on the ability of a metric to distinguish good candidate utterances from bad ones. As such, we need utterances that are clear and consistent in their quality. To ensure consistent quality, we rely on humans to generate our candidate utterances. The utterances we solicited each fall into one of three categories: *descriptive*, *ambiguous*, or *misleading*:

1. Descriptive Candidates: These consist of informative descriptions that are intended to

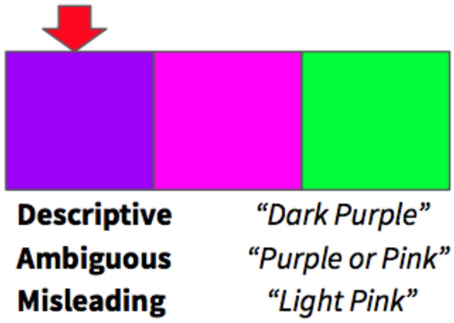


Figure 1: A hypothetical context with captions of different qualities. The red arrow points to the target color. The descriptive caption picks out the target, the ambiguous one selects two colors, and the misleading caption picks out a distractor color.

distinguish between the target and the distractors (i.e. non-target) colors. These should receive the highest scores.

2. Ambiguous Candidates: These consist of uninformative descriptions that are intended to correctly describe the target and at least one of the distractors. These should receive scores in the middle of the scale.

3. Misleading Candidates: These consist of descriptions that are intended to describe one of the distractors and not the target. These should receive the lowest scores.

We obtained our *descriptive* candidates by augmenting the dataset of Monroe et al. (2017). We selected 360 distinct color context–utterance pairs from the development set in which listeners were able to correctly identify the target. Because these metrics perform better with more references, we collected 5 reference descriptions for each of the 360 contexts from Mechanical Turk workers. Instead of having workers play the reference game in pairs, we described the game and asked that they play the speaker role. Separately, we then had three crowdworkers perform the listener role with each utterance. We kept only the utterances where at least two of the listeners identified the target correctly. We ended up with a total of 1,912 descriptive candidates with on average 5.2 references per context.

The Monroe et al. (2017) dataset does not contain labeled ambiguous descriptions, so we obtained our *ambiguous* candidates by having Mechanical Turk workers play the color reference game in the 360 color contexts as "ambiguous" speakers. The ambiguous speakers were asked to provide a description that applied to the target color while making it difficult for the listener to select the target. Any ambiguous descriptions that matched descriptive candidates for their context exactly were discarded. Some examples include "Blue" when the context contains a dark blue target and a light blue distractor, or "Color of the rainbow". Whether these captions are ambiguous in the sense that they communicate no relevant information or merely underspecified in the sense that that they do not provide enough information, these captions are of lower quality than the descriptive ones. There are 1,343 ambiguous candidates.

Finally, the Monroe et al. (2017) corpus does not explicitly contain misleading descriptions, but we did obtain a portion of our *misleading* candidates from their dataset. To do so, we made sure to select our 360 contexts as 180 context pairs. Each pair contains the same colors, but with a different target color. Therefore, a descriptive candidate for one context in the pair is a misleading caption for the other—the description directs the listener to the wrong color. Descriptive candidates from contexts with the same colors but different targets are our misleading candidates. We expect that these misleading candidates should be different from the descriptive ones, but they may be the same if all the colors are similar. To ensure that the descriptive and misleading candidates were distinct in the cases where the colors were different, we removed misleading candidates found in their context's reference sets if the distance between colors had a distance of at least 20 according to the CIEDE2000 standard (Sharma et al., 2005). There are 1,909 misleading candidates in all.

Because descriptive candidates pick out the target color, they are better than ambiguous candidates, and because ambiguous candidates apply to the target color, they are better than misleading candidates. An effective evaluation metric should then assign the highest scores to descriptive candidates, middle-of-the-range scores to ambiguous candidates, and the lowest scores to misleading candidates. An example of what these captions might look like can be found in Figure 1.

Our dataset and code can be found at `https://github.com/bnewm0609/comm-eval`.

5.2 Models for Communication-based Evaluation

To use our communication-based evaluation method in the color reference game scenario, we need to define our world states, speaker distributions, and listener distributions. The set of world states W_t includes one state in which each color in the context is the target, and the speaker's observed distribution $S(w_t)$ puts all its probability mass on the true target color. The listener's distribution $L(w_t \mid u)$ requires further consideration. This distribution can be modeled as any distribution over the world states conditioned on an utterance. We introduce three ways to generate such a distribution: human listeners, a Literal Listener model, and a Pragmatic Listener model.

To obtain a human listener in the sense of our evaluation, we had Mechanical Turk workers play the role of listeners in the reference game: they were given color contexts and candidate descriptions from each of the quality categories and were asked to select the color that the candidate best describes. The distribution they represent, $L(w_t \mid u)$, has all of its probability mass on the color they select. We had three workers play the reference game with each candidate utterance we collect.

If human data is unavailable, the distribution $L(w_t \mid u)$ can be modeled computationally. We consider two such models.

The first model is a "Literal Listener". The model takes an utterance as input and uses it to directly compute a distribution over world states. Following Monroe et al. (2017), we parameterize this Literal Listener with an LSTM that produces a mean color vector μ and covariance matrix Σ from an utterance, and these are used to score each context color f:

$$\text{score}(f) = -(f - \mu)\Sigma(f - \mu) \tag{3}$$

The scores are then normalized using a softmax function to obtain the required distribution over colors representing $L(w_t \mid u)$. We trained our model on the ≈15,000 utterances in the training set specified by Monroe et al. (2017), and evaluated on the test set of approximately the same size. We found that the target is assigned the highest score 76.53% of the time, much higher than chance performance of 33%.

In contrast to our Literal Listener model, our "Pragmatic Listener" model finds the probability of the *candidate utterance* given that each color in the context is the target, $P(u \mid w_t)$. These probabilities are used to derive $L(w_t \mid u)$ using Bayes' rule. To find the probability of the utterance, we use an LSTM as a conditional language model. The model is trained and structured following Monroe et al. (2017), and initialized with pretrained GloVe embeddings (Pennington et al., 2014). Inverting with Bayes' rule involves specifying a prior over utterances, and we treat this prior as uniform for simplicity. This model is *pragmatic* in the sense that it explicitly takes into account the view of a hypothetical speaker. This is the path taken in the automatic metrics used by Monroe et al. (2018) and Cohn-Gordon et al. (2018). This listener assigns the highest score to the target color in 75.02% of test-set contexts, also much better than chance.

Finally, we use our listener and speaker distributions to assign a score to the utterance following (1). Because the speaker's distribution has all probability on one world state, the score $M(u \mid L, S)$ reduces to the negative log likelihood of the target world state w_{target} given the utterance, as in (2) above. To put this score into a space similar to the F-measure spaces of the n-gram overlap metrics, we report $e^{-M(u|L,S)}$, or equivalently, the listener's probability for the target color.

It is important to note that because the states are defined only in terms of the target color, the only aspect that matters to an utterance's quality is whether it leads a listener to select the target. We do not explicitly evaluate stylistic aspects such as grammaticality or politeness, though the world states and distributions could be augmented to include these as in Kao et al. 2014. By defining our task in this manner, we are assuming that stylistic elements do not contribute to communicative success. While this is certainly not true in many situations, we believe is appropriate for this particular context.

5.3 Comparisons

To evaluate the effectiveness of our metrics at detecting how well utterances communicate a speaker's beliefs, we investigate the extent to which good utterances receive high scores and bad utterances receive low scores. We evaluate all of the utterances in each of the three quality categories: descriptive, ambiguous, and misleading.

First, for our baseline experiments, we run the n-gram overlap metrics to compare each of the de-

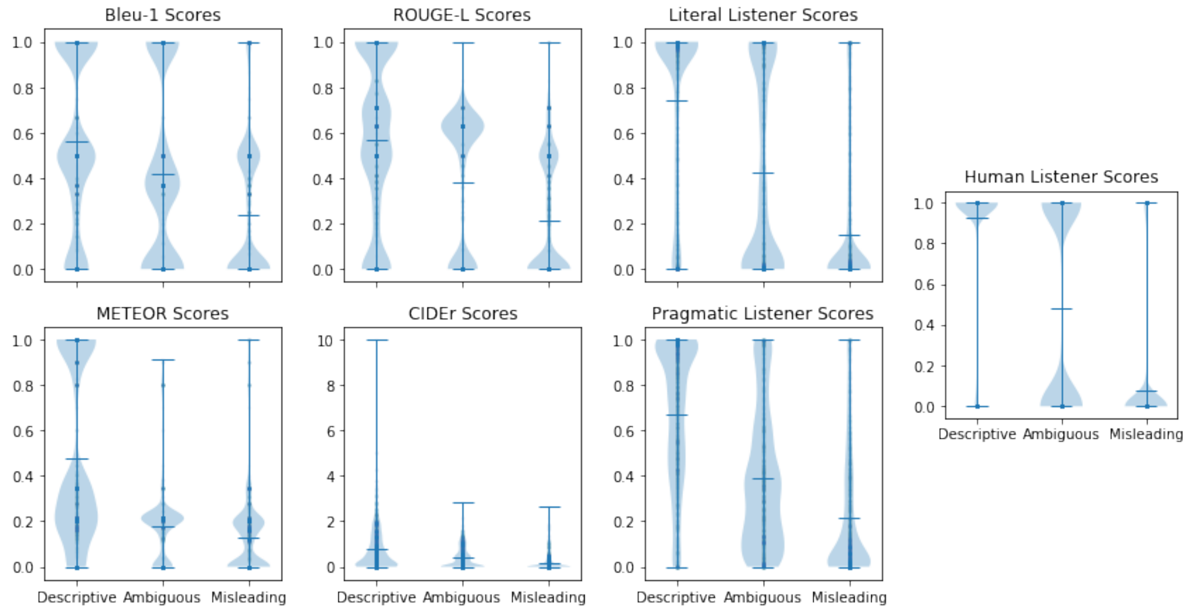


Figure 2: Violin plots showing the distribution of scores assigned by each metric across the three caption qualities. In the first two columns, we have the four n-gram overlap baselines. In the third column, we have the listener model metrics. On the right, we have the gold-standard human listener results. Violin plots are created with Gaussian kernel density estimate with bandwidth 0.2. Horizontal bars show ranges and means.

Metric	ρ	r	τ
Human	0.701	0.701	0.661
Literal Listener	0.581	0.613	0.486
Pragmatic Listener	0.554	0.556	0.444
BLEU-1	0.363	0.350	0.290
ROUGE-L	0.441	0.439	0.378
METEOR	0.482	0.479	0.404
CIDEr	0.401	0.417	0.340

Table 1: Pearson's ρ, Spearman's r, and Kendall's τ correlation values between assigned scores and quality categories. BLEU-1 is reported because it is the best of the BLEU scores. All correlations are significant at $p < 0.05$ and all Pearson's correlations are different at $p < 0.05$ according to a Williams' test.

scriptive, ambiguous, and misleading candidates to references from their contexts. We run these assessments with the `nlgeval` package (Sharma et al., 2017). We report the smoothed distributions of n-gram overlap scores for each category separately in the left two columns of Figure 2.

Next, we run our two communication-based evaluation models on each of these candidates. The score reported for an utterance is the probability the model assigns to the true target color being the target after processing the utterance. Again, we report the distribution of scores separated by category in the third column of Figure 2.

Finally, we plot our ground-truth human-listener scores. If the human listener correctly identified the target, the caption they saw received a score of 1; if they did not, the caption received a score of 0. Because we asked three crowdworkers to play the role of the listener for the captions we collected, we have 4,353 scores for descriptive captions, 4,029 for ambiguous captions, and 4,353 for misleading captions. The smoothed distribution of scores is on the far right in Figure 2.

We want to see the extent to which these scores correlate with the quality categories of the given utterances. Following the logic of Section 5.1, we assign descriptive candidates a score of 1, ambiguous candidates a score of 2, and misleading candidates a score of 3, and we report correlations calculated by Pearson's ρ, Spearman's r, and Kendall's τ. In this situation, we have large numbers of points around certain scores (e.g. 1 for the Literal Listener), and these scores have meaning themselves, so we report Pearson's ρ. We are also interested in the overall monotonicity of the metric scores across categories—we want to avoid good candidates receiving bad scores and vice versa. As such, we report the Spearman's r and Kendall's τ as well. The magnitudes of these coefficients are in Table 1.

We are also interested in the extent to which our method's correlations differ from the n-gram overlap ones, so we run a Williams' test for dependent Pearson's correlations. We find all Pearson's correlations are significantly different at $p < 0.05$.

6 Discussion

6.1 Qualitative Analysis

The results we observe are in accordance with the widely attested observation that n-gram overlap metrics do not capture human judgments particularly well (Novikova et al., 2017; Kilickaya et al., 2017). While all of the correlations are relatively weak, METEOR is the strongest n-gram overlap metric—its use of synonyms may very well aid it in this color-reference scenario. The success of it and ROUGE-L compared to other metrics points to recall being an important component of informativity in this task. This makes sense: if a candidate utterance does not contain enough of the n-grams found in a reference, it will likely be more difficult for a listener to select the target. On the other end, BLEU has the worst correlation. Additionally, metrics like ROUGE-L, BLEU, and CIDEr have been shown to correlate with human judgments on a system rather than individual sentence level (Novikova et al., 2017). Our results corroborate this poor sentence-level performance.

Previous work has found that n-gram overlap metrics are able to assign low scores to poorly judged utterances but fail to assign high scores to positively judged ones (Chaganty et al., 2018; Novikova et al., 2017). Our results provide some support for this claim, especially for METEOR and CIDEr. BLEU and ROUGE-L, however, give mid-to-high scores to a large number of utterances regardless of their quality.

Finally, it is clear that human listeners are performing a reasonable evaluation, tightly aligned with the quality categories. We also observe that the human listener score distribution is closely mirrored by the Literal Listeners' scores. However, the bimodal nature of the scores given to ambiguous sentences is not ideal. We seek a metric that assigns ambiguous utterances mid-range scores to reflect that they convey some information, but these are rare in the human responses and model predictions. Despite this, the superiority of the listener methods over the n-gram methods is evident both in the shapes of the distributions and their correlations.

6.2 Literal vs. Pragmatic Listener

Even though the Literal and Pragmatic Listener models are more effective than n-gram overlap metrics, they do evaluate the descriptive and ambiguous candidates differently. As noted above, the Literal Listener seems to work in a very polarized manner: captions are either good, earning a high score, or bad, earning a low score, without much in between. This is likely a result of training the Literal Listener model with a cross-entropy loss objective. This training scheme does not reward high-entropy distributions over outputs and pushes the model to always output a confident score (closer to one). This problem is not quite as apparent with the Pragmatic Listener, but many of the descriptive and ambiguous candidates appear to be assigned a range of higher scores. Interestingly, the Pragmatic Listener's distributions have higher entropy than the Literal Listener's. This might be because the Pragmatic Listener is based on a language model, so the probabilities it assigns reflect the probabilities of potentially multiple tokens. Some might be less informative than others, which would smooth out the distribution over colors. All told, the Literal Listener correlates slightly better with the quality categories than the Pragmatic Listener does.

6.3 Quality of the Listener Model

If our communication-based method is to be effective, the listener model used must be accurate. This is because our evaluation method assumes that communicative errors are the fault of the speaker and not the listener. Realistically, this is not the case—no listener, human or model, is perfect. Although our listener models are not 100% accurate, they are still able to distinguish between candidates of different qualities. In other words, despite their imperfections, these models are still reliable evaluators.

6.4 Shortcomings of Communication-based Evaluation

Hashimoto et al. (2019) claim that a sufficient evaluation method will incorporate the "quality" of a model's utterances as well as its "diversity". Quality is tied to precision—a good model's utterances are effective. Diversity is tied to recall—a good model will be able to produce any utterance a human might. Our method focuses solely on the quality aspect of this picture. To see why this may be problematic, note that a system that simply looked up descriptions in our data given contexts would appear perfect despite not meeting any diversity goals. This means that, if we want to measure diversity, we have to resort to a second metric (e.g. perplexity or HUSE-D; Hashimoto et al. 2019). That said, current automatic measures of quality, like n-gram overlap metrics, are not effective, and our proposed method addresses this.

Another caution is that our method depends only on the communicative goal of the speaker, which reduces the importance of other aspects of utterance quality. For example, in our color reference game scenario, grammaticality of utterances is only evaluated to the extent that grammatical descriptions aid a listener in selecting the correct color. If "blue dark on click the" and "click on the dark blue" both lead to the listener selecting the dark blue color, they will both be regarded as equally good, even though only the second is well-formed. Evaluating other aspects of quality, such as politeness, style, or tone, similarly requires careful consideration. Each of these can be thought of as achieving some communicative goal, but this goal along with the listener models and world states must be specified carefully to ensure that such properties are taken into account.

7 Conclusion

We developed an NLG evaluation method that is motivated by the idea that an utterance's quality is determined by how well it leads a listener to accurately recover the speaker's communicative intentions. We evaluated the effectiveness of this evaluation method using a simple color reference game in which we could systematically vary utterance quality and then assess how well different methods correlate with quality in this sense. In this setting, our communication-based method dramatically out-performed standard n-gram-based methods. What's more, our method can be used in any setting in which there is a well-defined action for a listener to perform in response to an utterance. One could, for example, apply this evaluation method to summarization, image captioning, translation, and even pure text generation, with tasks such as recovering the input from distractors, identifying salient points or features, or capturing shades of meaning. Although our method arguably does not capture every sense of quality that we might have for NLG, it does key directly into a fundamental goal we have for these systems, which is that they communicate effectively with humans using natural language.

8 Acknowledgements

The authors would like to thank Julia Gong and Suvir Mirchandani for their early assistance as well as the Stanford CSLI 2019 research interns, particularly Hanson Lu and Josephine Soddano. Thank you also to our reviewers for their helpful comments, and to our Amazon Mechanical Turk participants for their invaluable work. This material is based in part upon work supported by the NSF under Grant No. SMA-1659585.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

Jacob Andreas, Anca Dragan, and Dan Klein. 2017. Translating neuralese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 232–242, Vancouver, Canada. Association for Computational Linguistics.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Elia Bruni and Raquel Fernández. 2017. Adversarial evaluation for open-domain dialogue generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–288, Saarbrücken, Germany. Association for Computational Linguistics.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2017. Referenceless quality estimation for natural language generation. *Proceedings of the 1st Workshop on Learning to Generate Natural Language*.

Andrew Finch, Yasuhiro Akiba, and Eiichiro Sumita. 2004. How does automatic machine translation evaluation correlate with human scoring as the number of reference translations increases? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, Cambridge, MA. Association for Computational Linguistics.

Noah Goodman and Michael Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Anjuli Kannan and Oriol Vinyals. 2016. Adversarial evaluation of dialogue models. *NIPS Workshop on Adversarial Training*.

Justine T Kao, Jean Y Wu, Leon Bergen, and Noah D Goodman. 2014. Nonliteral understanding of number words. *Proceedings of the National Academy of Sciences*, 111(33):12002–12007.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic Turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126, Vancouver, Canada. Association for Computational Linguistics.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Will Monroe, Jennifer Hu, Andrew Jong, and Christopher Potts. 2018. Generating bilingual pragmatic color references. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2155–2165, New Orleans, Louisiana. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Gaurav Sharma, Wencheng Wu, and Edul N Dalal. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 30(1):21–30.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. *CoRR*, abs/1701.02870.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D. Manning, and Curtis P. Langlotz. 2018. Learning to summarize radiology findings. In *EMNLP 2018 Workshop on Health Text Mining and Information Analysis*.

Multi-Input Strictly Local Functions for Tonal Phonology

Jonathan Rawski and Hossep Dolatian
Department of Linguistics
Institute for Advanced Computational Science
Stony Brook University
{jonathan.rawski,hossep.dolatian}@stonybrook.edu

Abstract

This paper presents an automata-theoretic characterization of the typology of attested tonal patterns using enriched data structures. We generalize the Input Strictly Local class of functions to consider multiple inputs of tonal and segmental strings, and find that the associated strictly local multi-tape transducers successfully capture tonal typology. Links between automata-theoretic and logical characterizations of phonological expressivity showcase tradeoffs in data structure and locality in the expressivity of phonological computation.

1 Introduction

Recent work in mathematical phonology connects phonological mappings to subclasses of the regular functions (McNaughton and Papert, 1971; Rogers and Pullum, 2011; Rogers et al., 2013; Heinz and Lai, 2013; Chandlee, 2014). One of the simplest subclasses is the class of Input Strictly Local (ISL) functions which take as input a single string *and* generate an output based on local information. Despite their reduced expressivity, ISL functions capture a majority of phonological and morphological maps (Chandlee, 2017; Chandlee and Heinz, 2018). In addition, ISL functions are provably easier and faster to learn than full regular functions (Chandlee et al., 2015a).

In this paper, we generalize this notion of locality from the above single-input functions to functions which take *multiple* strings as input in §2. Such functions are *Multi-Input Strictly Local* (MISL). MISL functions are effectively computed by a class of deterministic asynchronous Multi-tape Finite State Transducers (MT-FSTs). Natural language has processes which are understood in terms of enriched multi-string input structures, i.e. autosegmental structure. We focus on tone association §3.

The bulk of computational results on tonal patterns are defined over graphical structures and are *local* over *autosegmental* graphs (Jardine, 2016a,b, 2017a, 2019; Chandlee and Jardine, 2019a). In §4, we show that the bulk of tonal processes are MISL: they are local when computed as a multi-input function over strings. This provides a solution to a dichotomy in formal language results between the complexity of segmental vs tonal phonology (Jardine, 2016a) via enriching the data structure in a linguistically natural way. This also connects logically defined functions to automata-theoretic characterizations over enriched data structures.

Tonal processes is sufficiently computable using types of MT-FSTs, but we show that the full power is not necessary. Showing that the bulk of tonal phonology can be computed with *only* MISL MT-FSTs, acts as a stepping stone to determining the learnability of tone. It likewise acts as a benchmark to examine the typology of attested and unattested tonal processes. Furthermore, by using multi-input functions with MT-FSTs instead single-input functions with FSTs, we can more iconically compute the fact that 1) the tone tier is separate from the vowel tier, and that 2) this separation makes certain tonal processes be local.

We emphasize that our result is NOT an argument against the use of graphs in tone. The use of graphs iconically captures tonal processes. Any linear encoding of autosegmental structure, including ours, requires the use of special symbols for preassociation (Kornai, 1995; Wiebe, 1992; Yli-Jyrä, 2013, 2015).

Single-input functions are a special case of multi-input functions. With finite-state calculus, single-input functions correspond to rational functions when modeled by 1-way single-tape FSTs, and to regular functions when modeled by 2-

way single-tape FSTs (Filiot and Reynier, 2016).[1] Multi-input functions are modeled by 1-way or 2-way MT-FSTs. Although there is work on the expressivity of multi-tape automata (Furia, 2012), little is known on multi-input *functions* and their algebra or expressivity (Frougny and Sakarovitch, 1993). We show that the MISL class characterizes a substantial chunk of tonal phonology.

2 Preliminaries

2.1 Preliminaries for single-input functions

Let $\rtimes$,$\ltimes$be the start and end boundaries respectively. Let Σ be a finite alphabet of symbols (excluding $\rtimes$,$\ltimes$). Let $\Sigma_{\rtimes} = \Sigma \cup \{\rtimes, \ltimes\}$. Let Σ^* the set of all strings over Σ. Let $|w|$ indicate the length of $w \in \Sigma^*$. For two strings w and v let wv be their concatenation, and for a set $L \subset \Sigma^*$ of strings and a string w, by wL we denote $\{wv|v \in L\}$. Let λ denote the empty string.

Given some string u and a natural number k, the *k-suffix* of u is the last k symbols of u: $\mathtt{suff}(u, k) = v$ s.t. $|v| = k$ and $xv = u$ for some $x \in \Sigma^*$. For an alphabet Σ, the *k-factors* of Σ are the set of strings $w \in \Sigma*$ such that $|w| \leq k$.

Informally, a single-input function f is k-ISL if for all $u_1, u_2 \in \Sigma^*$, if $\mathtt{suff}(u_1, k-1) = \mathtt{suff}(u_2, k-1)$ then the two strings have the output extensions w.r.t f (Chandlee, 2014; Chandlee et al., 2015b). For any k-ISL function f over domain Σ^*, there exists a *canonical* deterministic single-tape finite-state transducer (1T-FST) M such that $|M| = f$ (meaning M computes f), and every state $q \in Q$ in M is labelled with one of the $k-1$ suffixes of Σ^*. Transitions are function tuples $\Delta : Q \times \Sigma \rightarrow Q \times \Gamma^*$. For a state $q \in Q$ and input symbol $a \in \Sigma$, $\delta(q, a) = (p, B)$ such that $B \in \Gamma^*$ and $p = \text{suff}(qa,)$.

2.2 Preliminaries for multi-input functions

We introduce notation for functions which take multiple strings as input. To do so, we use tuples demarcated by brackets. In the formalization here, we only consider functions which produce one output string, not a tuple of output strings. But extending the formalization is trivial; such a function is illustrated in §4.3.1.

[1] By single-tape FST, we mean a two-tape MTFST with one input tape and one output tape. Note that the functions computed by 1-way FSTs are called ‘regular functions’ in American computer science. In this paper, we follow French conventions which call this class the ‘rational functions’ (Filiot and Reynier, 2016).

A function f is an n-input function if it takes as input a tuple of n strings: $[w_1, \ldots, w_n]$, which we represent as $\vec{w}$, where each word w_i is made up of symbols from some alphabet Σ_i such that $w_i \in \Sigma_i^*$. Each alphabet Σ_i may be disjoint or intersecting, so two input strings w_i, w_j may be part of the same language Σ_i^*. These n alphabets form a tuple $\vec{\Sigma}$. Tuples can be concatenated: if $\vec{w} = [ab, c], \vec{x} = [d, ef]$, then $\vec{w}\vec{x} = [abd, cef]$.

To generalize the notion of suffixes into multiple strings, we define a tuple of n natural numbers as $\vec{k} = [k_1, \ldots, k_n]$. Given some tuple of n strings $\vec{w}$ and tuple of n numbers $\vec{k}$, *$\vec{k}$-suffix* of $\vec{w}$ is a tuple $\vec{v}$ of n strings v_i, made up of the last k_i symbols on w_i: $\mathtt{suff}(\vec{w}, \vec{k}) = V$ s.t. $\vec{v} = [v_1, \ldots, v_n]$ and $|v_i| = k_i$ and $x_i v_i = w_i$ for $x_i \in \Sigma_i^*$. E.g. for $\vec{w}$=[abc,def] and $\vec{k} = [2, 1]$, $\mathtt{suff}(\vec{w}, \vec{k}) = [bc, f]$. Given a tuple $\vec{k}$, the operation $\vec{k} - x$ subtracts x from each of k_i. E.g., for $\vec{k} = [2, 3, 6]$, $\vec{k} - 1 = [1, 2, 5]$. For a tuple of alphabets $\vec{\Sigma}$, the $\vec{k} - factors$ of $\vec{\Sigma}$ is the set of tuples $\vec{w} \in \vec{\Sigma}$ such that $|w_i| \leq k_i$.

Let f be an n-input function defined over an $n-$tuple $\vec{w}$ of input strings $\vec{w} = [w_1, \ldots, w_n]$ taken from the tuple of n alphabets $\vec{\Sigma}$. As an *informal* and intuitive abstraction from ISL functions, f is Multi-Input Strictly Local (MISL) for $k = [k_1, \ldots, k_n]$ if the function operates over a bounded window of size k_i for w_i. Formally,

Definition 1: A function f is $\vec{k}$-MISL iff there exists a deterministic asynchronous Multi-tape FST such that i) $|M| = f$, and ii) the MT-FST is canonically $\vec{k}$-MISL

We explain $\vec{k}$-MISL Multi-tape FSTs in the next section.

Note that Definition 1 is an automata-theoretic definition, meaning the expressivity is necessarily dependent on the machine. A language-theoretic definition of MISL functions, and connections to this class of multi-tape transducers, is in progress. While ISL FSTs and MISL MT-FSTs similarly encode the k-suffix information and the notion of common output in the state of the transducer, the use of common output extensions used in the ISL functions is not easily extendable to multi-input functions. In particular, there are non-subsequential n-input functions which are computable with MISL MT-FSTs.

For an ISL function, it does not matter if the input string is read left-to-right or right-to-left. But for an MISL function, it does. A function may be left-to-right MISL but not right-to-left MISL. We leave out a proof but an illustration is given in §4.1.

2.3 Multi-tape finite-state transducers

Multi-input functions can be modeled by multi-tape FSTs (MT-FST). An MT-FST is conceptually the same as single-tape FSTs, but over *multiple* input tapes (Rabin and Scott, 1959; Elgot and Mezei, 1965; Fischer, 1965; Fischer and Rosenberg, 1968; Furia, 2012). MT-FSAs and MT-FSTs are equivalent, and single-tape FSTs correspond to an MT-FSA with two tapes.

Informally, a MT-FST reads n multiple input strings as n input tapes, and it writes on a single output tape. Each of the n input strings is drawn from its own alphabet Σ_i. The output string is taken from the output alphabet Γ. For an input tuple of n strings $\vec{w} = [w_1, \dots, w_n] = [\sigma_{1,1} \dots \sigma_{1,|w_1|}, \dots, \sigma_{n,1} \dots \sigma_{n,|w_n|}]$, the initial configuration is that the MT-FST is in the initial state q_0, the read head. The FST begins at the first position of each of the n input tapes $\sigma_{i,1}$, and the writing head of the FST is positioned at the beginning of an empty output tape. After the FST reads the symbol under the read head, three things occur: 1) the state changes; 2) the FST writes some string; 3) the read head may advance to the right (+1) or stay put (0) on different tapes: either move on all tapes, no tapes, or some subset of the tapes.

This process repeats until the read head "falls off" the end of each input tape. If for some input $\vec{w}$, the MT-FST falls off the right edge of the n input tapes when the FST is in an accepting state after writing u on the output tape, we say the MT-FST transduces, transforms, or maps, $\vec{w}$ to u or $f_T\vec{w} = u$.[2] Otherwise, the MT-FST is undefined at $\vec{w}$. We illustrate MT-FSTs in §4.

A $n-$MT-FST is a 6-tuple $(Q, \vec{\Sigma_{\rtimes}}, \Gamma, q_0, F, \Delta)$ where:

- $n \in \mathbb{N}$ is the number of input tapes
- Q is the set of states
- $\vec{\Sigma_{\rtimes}} = [\Sigma_{1\rtimes}, \dots, \Sigma_{n\rtimes}]$ is a tuple of n input alphabets Σ_i which include the end boundaries $\Sigma_{i\rtimes}$
- Γ is the output alphabet
- $q_0 \in Q$ is the initial state
- $F \subset Q$ is the set of final states
- $\delta : Q \times \vec{\Sigma_{\rtimes}} \rightarrow Q \times \vec{D} \times \Gamma^*$ is the transition function where
  - $D = \{0, +1\}$ is the set of possible directions,[3]
  - $\vec{D} = [D^n]$ is an n-tuple of possible directions to take on each tape

The above definition can be generalized for MT-FSTs which use multiple output tapes. As parameters, an MT-FST can be deterministic or non-deterministic, synchronous or asynchronous. We only use *deterministic* MT-FSTs which are weaker than non-deterministic MT-FSTs. An MT-FST is synchronous if all the input tapes are advanced at the same time, otherwise it is asynchronous. We use asynchronous MT-FSTs which are more powerful than synchronous MT-FSTs. Synchronous MT-FSTs are equivalent to multi-track FSAs which are equivalent to single-tape FSAs, making them no more expressive than regular languages. For a survey of the properties of MT-FSAs and MT-FSTs, see Furia (2012).

A configuration c of a $n-$MT-FST M is an element of $(\vec{\Sigma_{\rtimes}}^* Q \vec{\Sigma_{\rtimes}}^* \times \Gamma^*)$, short for $([\Sigma^*_{1\rtimes} q \Sigma^*_{1\rtimes}, \dots, \Sigma^*_{n\rtimes} q \Sigma^*_{n\rtimes}] \times \Gamma^*)$. The meaning of the configuration $c = ([w_1 q x_1, \dots, w_n q x_n], u)$ is the following. The input to M is the tuple $\vec{w}\vec{x} = [w_1 x_1, \dots, w_n x_n]$. The machine is currently in state q. The read head is on each of the n-input tapes on the first symbol of x_i (or has fallen off the right edge of the input tape if $x_i = \lambda$). u is currently written on the output tape.

Let the current configuration be $([w_1 q a_1 x_1, \dots, w_n q a_n x_n], u)$ and let the current transition arc be $\delta(q, [a_1, \dots, a_n]) = (r, \vec{D}, v)$. If $\vec{D} = [0^n]$, then the next configuration is $([w_1 r a_1 x_1, \dots, w_n r a_n x_n], uv)$ in which case we write $([w_1 q a_1 x_1, \dots, w_n q a_n x_n], u) \rightarrow ([w_1 r a_1 x_1, \dots, w_n r a_n x_n], uv)$ (= none of the tapes are advanced) . If $\vec{D} = [+1^n]$, then the next configuration is $([w_1 a_1 r x_1, \dots, w_n a_n r x_n], uv)$ in which case we write $([w_1 q a_1 x_1, \dots, w_n q a_n x_n], u) \rightarrow ([w_1 a_1 r x_1, \dots, w_n a_n r x_n], uv)$ (= all the tapes are advanced). Otherwise, the next configuration is $([w_i C_1 x_1 \dots, w_n C_n x_n, \dots], uv)$ where $C_i = r a_i$ if $D_i = 0$ and $C_i = a_i r$ if $D_i = +1$ in which case we write $([w_1 q a_1 x_1, \dots, w_n q a_n x_n], u) \rightarrow ([w_i C_1 x_1 \dots, w_n C x_n, \dots], uv)$ (= a subset of the tapes are advanced).[4]

[2]If the MT-FST generates tuples instead of single strings, then the MT-ST maps $\vec{w}$ to $\vec{u}$.

[3]If the MT-FST reads from right to left, then it uses the -1 direction parameter

[4]Note that the interpretation of the third type of configuration subsumes the first two. We explicitly show the first two

The transitive closure of $\rightarrow$ is denoted with $\rightarrow^+$. Thus, if $c \rightarrow^+ c'$ then there exists a finite sequence of configurations $c_1, c_2 \ldots, c_n$ with $n > 1$ such that $c = c_1 \rightarrow c_2 \rightarrow \ldots \rightarrow c_n = c'$.

As for the function that a MT-FST M computes, for each $n-$tuple $\vec{w} \in \vec{\Sigma}^*$ where $\vec{w} = [w_1, \ldots, w_n]$, $f_M(\vec{w}) = u \in \Gamma^*$ (where $f_M = |M|$) provided there exists $q_f \in F$ such that $([q_0 \rtimes w_1 \ltimes, \ldots, q_0 \rtimes w_n \ltimes], \lambda) \rightarrow^+ ([\rtimes w_1 \ltimes q_f, \ldots, \rtimes w_n \ltimes q_f], u)$. Otherwise, if the configuration is $([\rtimes w_1 \ltimes q, \ldots, \rtimes w_n \ltimes q], u)$ and $q \notin F$ then the transducer crashes and the transduction f_T is undefined on input $\vec{w}$. Note that if a MT-FST is deterministic, it follows that if $f_T(\vec{w})$ is defined then u is unique.

As explained in §2.2, we define a function as $\vec{k}$-MISL iff there exists a corresponding deterministic asynchronous $\vec{k}$-MISL Multi-tape FST.

Definition 2: A deterministic asynchronous MT-FST M with alphabet $\vec{\Sigma}$ is a canonical MT-FST for an $\vec{k}$-MISL function f if the states of M are labelled with the $\vec{k}-1$ suffixes of $\vec{\Sigma}$.

In Definition 2, the restriction on state labels does not apply to the unique initial state and unique final state. In other words, except for the initial and final states q_0 and q_f, every state corresponds to a possible $\vec{k}-1$ *factor* of f

.

3 Computational phonology of tone

Segmental phonological processes are generally computed as single-input functions and they are ISL (Chandlee, 2014; Chandlee and Heinz, 2018). But when treated as a single-input function, tonal processes are significantly more complex than ISL (Jardine, 2016a). Single strings also fail to capture the suprasegmental nature of tone. Instead, tonal processes are generally modeled with *autosegmental representations* (ASR). As graphs, ASRs are a richer data structure that showcase the non-linear nature of tone by breaking up a linear string into parallel strings or tiers (tone and vowel/mora).

As a review, consider the nonce words in Table 1. On the surface, the vowels each surface with some tone feature: high V́ vs. low V̀. A common analysis is that underlyingly the tones are on a separate tier from the vowels. A mapping function creates association arcs between the tones and vowels. In the input in Table 1a, then the tones and vowels are not underlying preassociated. Some tonal processes are analyzed with underlying preassociated tones (Table 1b). That is, the input contains an association arc between the some of the tones and some of the vowels.

Most mathematical results on tonal phonology are also defined over graphs or graph-like structures (Bird and Klein, 1990; Bird, 1995; Coleman and Local, 1991; Coleman, 1998). Jardine (2016a,b, 2017a) showed that computing well-formedness for tonal structures is Strictly Local over ASRs. For transformations, Chandlee and Jardine (2019a) define a class of logical functions over ASRs called Autosegmental Input-Strictly Local functions (A-ISL), which can model many but not all tonal mappings that have preassociation. Informally, a function is A-ISL if it consists of two ISL functions operating over two tiers or two separate strings.[5] Koser et al. (2019) showed that mapping ASRs without preassociation to ASRs with associations is likewise a local process, specifically with Quantifier-Free Least Fixed Point logic (QFLFP) (Chandlee and Jardine, 2019b). However, most of these results are defined logically (Jardine, 2017b, 2019), and do not clearly correspond to other algebraic or automata-theoretic notions.

Computationally, tonal processes have been modeled with single-tape FSTs (Bird and Ellison, 1994; Kornai, 1995; Yli-Jyrä, 2013, 2015), synchronous MT-FSTs (Kiraz, 2001), and non-deterministic asynchronous MT-FSTs (Kay, 1987; Wiebe, 1992). To our knowledge, the above mathematical properties of tone as a *graph* have not been linked with finite-state calculus. As a link, we treat tonal processes as a *multi-input* function that takes as input a tuple of two strings. With this definition, the bulk of tonal processes are MISL.

4 Multi-Input Locality in Tone

Table 2 illustrates all the tonal functions which we formalize. Items a-e are taken from Koser et al. (2019), and items f-l from Chandlee and Jardine (2019a). Throughout this section, we reference only this table; see the original references for more language information.

Items a-e are not ISL but are A-ISL.[6] In §4.1, we show they are also MISL. Items f-l have preassociated tone-vowel pairs in the input. In §4.2, we

for illustrative reasons.

[5]There are much more nuances to the definition of A-ISL; readers are referred to Chandlee and Jardine (2019a).

[6]Koser et al. (2019) formalize tonal functions without preassocation with Quantifier-Free Least Fixed Point logic.

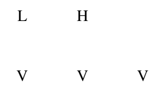
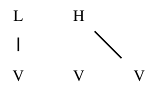
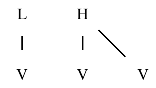
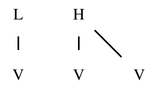
	a. Without underlying preassociation	b. With underlying preassocation
Input as string	*LH + patuki*	*patúki*
Input as graph		
Output as string	*pàtúkí*	*pàtúkí*
Output as graph		

Table 1: Review of tonal phonology.

show that with a specific linear encoding for preassociation, all the relatively simple ISL or A-ISL patterns are also MISL. More complex cases are handled in §4.3.

4.1 Tone without preassocation

4.1.1 General illustration: Mende spreading

We first illustrate with Mende (2a) which has a process of left-to-right tonal spread. Tones and vowels match 1-1 up until the last tone: *nìkílì* 'groundnut'. If there are more vowels than tones, then the final tone spreads: *félàmà* 'junction'.

As a function f, Mende left-to-right spreading is a 2-input function that takes as input a tuple of two strings: $\vec{w} = [w_1, w_2]$. The input string w_1 is a string of tones **T** taken from the input alphabet $\Sigma_1 = \Sigma_T = \{H, L\}$. The input string w_2 is a string of vowels **V** taken from the input alphabet $\Sigma_2 = \Sigma_V = \{V\}$. The input language is thus a tuple of two regular languages $[\Sigma_T^*, \Sigma_V^*]$. Each alphabet can include the start and end boundaries $\rtimes, \ltimes$: $\Sigma_{i\rtimes} = \Sigma_i \cup \{\rtimes, \ltimes\}$. The function generates a single output string of tonal vowels: $\Gamma = \{$V́,V̀$\}$.

This 2-input function is MISL for $\vec{k} = [2, 1]$. It needs a locality window of size 2 on the **T**-string in order to know if some tone is final or not (i.e., if we see $H\ltimes$ or $L\ltimes$), and a locality window of size 1 on the **V**-string because the function only needs to know the current vowel.

This function is computed by the deterministic asynchronous MT-FST in Figure (1). It uses two input tapes: a tone tape **T** and a vowel tape **V**. The MT-FST has a dedicated initial and final state q_0 and q_f. All other states are labelled with the $\vec{k}-1$-factors separated by commas. Transitions have the template $[\Sigma_1, \Sigma_2, \ldots, \Sigma_n]$: $[D^n]$: Γ^* where Σ_i

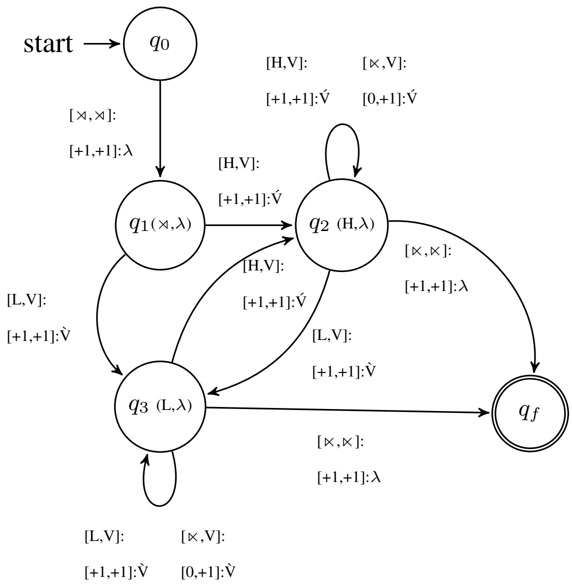


Figure 1: MT-FST for Mende

marks the read input symbols on the input string w_i, and where D is a possible direction parameter from {0,+1}. Given a parameter D_i, the transition arc dictates whether the MT-FST will advance (+1) or stay put (0) on the input tape w_i.

A sample derivation for /HL + felama/ is in Table 3. Each row keeps track of the: i) current state, ii) location of the read head on the input tapes, iii) transition arc used on each input tape, iv) outputted symbol, v) current output string. At step 5, upon reading $\ltimes$on the **T**-tape, asynchrony allows the read-head to advance on the **V**-tape but not on the **V**-tape, capturing the spreading effect.

4.1.2 Other processes without preassociation

Data in this section is illustrated in Table 2b-e and collected from Koser et al. (2019) who showed that they are are local in that they are QFLFP. We

Table 2: Sample of tonal processes, example input-output structures, and computational complexity.
Legend: * Function was proved to be QFLFP by Koser et al. (2019), ** Function is MISL if the output is 2-tuple

	Language	Process	Input		Output	Pre-ass?	ISL	A-ISL	MISL	$\vec{k}$-value
a	Mende	Iterative left-right spread	/LH + VVV/ L H V V V	→	[V̀V́V́] L H V V V	✗		✓*	✓	[2,1]
b	Kikuyu	Initial spread to two + final spread	/LHLH + VVVVVVV/ L H L H V V V V V V V	→	[V̀V̀V́V̀V́V́V́] L H L H V V V V V V V	✗		✓*	✓	[2,3]
c	Hausa	Iterative right-left spread	/LH + VVV/ L H V V V	→	[V̀V̀V́] L H V V V	✗		✓*	✓	[2,1]
d	Northern Shona	Edge-in + initial spread + medial spread	/HLH + VVVVVV/ H L H V V V V V V	→	[V́V́V́V̀V̀V́] H L H V V V V V V	✗		✓*	✓	[4,6]
e	Kukuya	Quantity sensitive spreading	/H + VVVV/ H V V V V	→	[V́V́V́V́] H V V V V	✗		✓*	✓	[4,2]
f	Rimi	Bounded tone shift	/VV́VV/ $/\langle\underline{H}\rangle + V\langle\underline{V}\rangle VV/$ H V V V V	→	[VVV́V] H V V V V	✓	✓	✓	✓	[1,2]
g	Zigula	Unbounded tone shift	/VVV́VVV/ $/\langle\underline{H}\rangle + V\langle\underline{V}\rangle VV/$ H V V V V V V	→	[VVVVV́V] H V V V V V V	✓	✗	✓	✓	[1,3]
h	Bemba	Bounded tone spread	/VV́VV/ $/\langle\underline{H}\rangle + V\langle\underline{V}\rangle VV/$ H V V V V	→	[VV́V́V] H V V V V	✓	✓	✓	✓	[1,2]
i	Arusa	Unbounded deletion	/V́ VV́V́V/ $/\langle\underline{H}\rangle\ \langle\underline{H}\rangle + \langle\underline{V}\rangle\ V(\underline{VV})V/$ H H V V V V V	→	[V́VVVV] H H V V V V V	✓	✗	✓	✓	[3,1]
j	Luganda	Bounded Meussen's rule	/V́V́V́V/ $/\langle\underline{H}\rangle\ \langle\underline{H}\rangle + \langle\underline{V}\rangle(\underline{VV})V/$ H H V V V V	→	[V́V̀V̀V] H L V V V V	✓	✓	✗	✓	[2,2]**
k	Shona	Alternating Meussen's rule	/V́-V́-V́/ $/\langle\underline{H}\rangle\text{-}\langle\underline{H}\rangle\text{-}\langle\underline{H}\rangle + \langle\underline{V}\rangle\text{-}\langle\underline{V}\rangle\text{-}\langle\underline{V}\rangle/$ H H H H V V V V V	→	[V́-V̀-V́] H L H L V V V V V	✓	✗	✗	✗	
l	Ndebele	Unbounded spreading to ante-penultimate	/V́VVVV/ $/\langle\underline{H}\rangle + \langle\underline{V}\rangle VVVV/$ H V V V V V	→	[V́V́V́VV] H V V V V V	✓	✗	✗	✓	[1,3]

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\underline{\rtimes}\mathrm{HL}\ltimes$	$\underline{\rtimes}\mathrm{eaa}\ltimes$		
2.	q_1	$\rtimes\underline{\mathrm{H}}\mathrm{L}\ltimes$ $\rtimes$:+1	$\rtimes\underline{\mathrm{e}}\mathrm{aa}\ltimes$ $\rtimes$:+1	λ	
3.	q_2	$\rtimes\mathrm{H}\underline{\mathrm{L}}\ltimes$ H:+1	$\rtimes\mathrm{e}\underline{\mathrm{a}}\mathrm{a}\ltimes$ e:+1	*é*	*é*
4.	q_3	$\rtimes\mathrm{HL}\underline{\ltimes}$ L:+1	$\rtimes\mathrm{ea}\underline{\mathrm{a}}\ltimes$ a:+1	*à*	*éà*
5.	q_3	$\rtimes\mathrm{HL}\underline{\ltimes}$ $\ltimes$:0	$\rtimes\mathrm{eaa}\underline{\ltimes}$ a:+1	*à*	*éàà*
6.	q_f	$\rtimes\mathrm{HL}\ltimes$ $\ltimes$:+1	$\rtimes\mathrm{eaa}\ltimes$ $\ltimes$:+1	λ	*éàà*

Table 3: Derivation of *HL* + *felama* over its tone-vowel tiers *HL* + *eaa* with the MT-FST in Figure 1

show that they are all MISL. Example MT-FSTs and derivations for cases b,c are in the appendix.

Kikuyu has a process of spreading an initial tone up to first two vowels (2b). The remaining tones and vowels are associated 1-to-1. If there are more vowels than tones, the final tone is spread: /LHLH + VVVVVVV/ → [V̀V̀ V́ V̀ V́V́V́]. Initial spreading up to two vowels is [2,3]-MISL because the function requires the context [$\rtimes$L,$\rtimes$VV] in order to spread L to the first two vowels. Final spread is [2,1]-MISL as in Mende (§4.1.1). Together, Kikuya is [2,3]-MISL.

Hausa (2c) behaves analogously to Mende but tones are associated *right-to-left* with *initial*-spreading: /LH + VVV/ → [V̀V̀ V́]. This is [2,1]-MISL *when* the input string is read right-to-left.

North Karanga Shona is more complex (2d). The initial and final tones are associated to the first and last vowels respectively. The first tone can spread up until the first 3 vowels *but not* to the penultimate vowel. The medial tone can spread up until the penultimate vowel: /HLH + VVVVVV/ → [V́V́V́ V̀V̀ V́]. The process is MISL but for a very large locality window of [4,6]. The window may be larger or smaller depending on various complications discussed in Koser et al. (2019).

Lastly, Kukuya (Table 2e) allows a H tone to spread if it is the only tone: /H + VVV/ → [V́V́V́]. Otherwise, if the input is HL, the L tone spreads: /HL + VVV/ → [V́ V̀V̀]. If LH, the L spreads up until the penultimate vowel: /LH + VVV/ → [V̀V̀ V́]. This is at most [4,2]-MISL: 4 over the **T**-tape in order to check if it's H, HL, or LH; 2 over the **V**-tape to prevent an L from spreading to the final vowel if the input tone is LH.[7]

[7]If the input tone is LHL, (Koser et al., 2019) do not state if either L can ever show spreading in words of four or more vowels. If they can, this is also MISL.

4.1.3 Contour tones

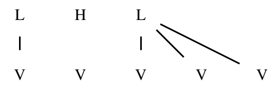
In §4.1, we assumed that the input had at least as many vowels as tones. If the input has more tones than vowels, final contour tones can be made: /HL + V/ → [V̂]. Assume that the number of possible contour tones is finite and modeled with a finite number of characters: rising V̌, falling V̂. To generate contour tones, one *compositional* approach is to first generate 1-to-1 or 1-to-many tone-vowel associations without any contour symbols; if there are more tones than vowels, then the unassigned tones are outputted at the end of the output string: /HL + V/ → //V́ L//. The string is then fed to an ISL function which changes strings of tonal vowels and tones into contour tones: //V́ L// → [V̂]. A non-compositional approach is mapping unassociated tones-and-vowels to the output through a single function. We conjecture that this function would be MISL as long as there are no long-distance dependencies involved in creating a contour tone. For easier illustration, we assume a compositional approach.

4.2 Tone with preassociation

4.2.1 Encoding preassociation

Tonal processes may include inputs where a tone is *preassociated* to one or more vowels. This dependency between the two strings is a reason why graphical structures are useful representations for tone, but it is a reason why many linear encodings require some special markup system (Kornai, 1995). For our purposes, we use the following encoding in Figure 2, inspired from an encoding system used by Yli-Jyrä (2013, 2015). We do not use other proposed encoding systems (Wiebe, 1992; Kornai, 1995; Yli-Jyrä, 2013, 2015) because they are either designed for single-tape FSTs or do not maintain strict locality.

If a tone T or single vowel V is preassociated, it is underlined and demarcated with angle brackets: $\langle\underline{\mathrm{T}}\rangle$, $\langle\underline{\mathrm{V}}\rangle$. If a span of multiple vowels are



Graph

Encoding $\langle\underline{\mathrm{L}}\rangle\mathrm{H}\langle\underline{\mathrm{L}}\rangle + \langle\underline{\mathrm{V}}\rangle\mathrm{V}(\underline{\mathrm{VVV}})$

Figure 2: Encoding preassociation

associated to the same tone, they are marked with parentheses instead of angle brackets: ($\underline{\mathrm{V}}$ $\underline{\mathrm{V}}$ . . . $\underline{\mathrm{V}}$). This encoding creates the following enriched input alphabets of multi-character units:

- $\Sigma_T = \{$ H, L, $\langle\underline{\mathrm{H}}\rangle$, $\langle\underline{\mathrm{L}}\rangle\}$
- $\Sigma_V = \{$ V, $\langle\underline{\mathrm{V}}\rangle$, ($\underline{\mathrm{V}}$, $\underline{\mathrm{V}}$, $\underline{\mathrm{V}}$) $\}$[8]

Other possible configurations, such as word-medial contour tones require a more elaborate encoding which we do not discuss. We set these aside because the preassociation data in Chandlee and Jardine (2019a) did not have such case studies.[9] We set aside the evaluation of our encoding mechanism based on Kornai (1995)'s desirada.

4.2.2 Locality of preassociated tones

With the above encoding, the tone functions in Table 2f-i with preassociation are MISL. Example MT-FSTs and derivations are in the appendix.

In Rimi (2f), a process of bounded tone shift will cause a preassociated tone to delink from its vowel and associate with the subsequent vowel: /V́VVV/→ [VVV́V]. In our encoding, the input is /$\langle\underline{\mathrm{H}}\rangle$ + V$\langle\underline{\mathrm{V}}\rangle$VV/. This function is ISL, MISL, *and* [1,2]-MISL. We need a locality window of size 1 over the **T**-string because we care if the current tone symbol is a preassociated $\langle\underline{\mathrm{H}}\rangle$. If yes, then we need a locality window of size 2 over the **V**-string in order to delink the current preassociated vowel $\langle\underline{\mathrm{V}}\rangle$ and associate the tone with the next vowel.

Unlike Rimi, Zigula displayed unbounded tone shift (2g) whereby a preassociated H is delinked from its preassociated vowel and associated with the *penultimate* vowel which can be at any distance away from the underlyingly preassociated vowel: / VVV́VVV/ or /$\langle\underline{\mathrm{H}}\rangle$ + VV$\langle\underline{\mathrm{V}}\rangle$VVV/ → [VVVVV́V]. This function isn't ISL but it is A-ISL *and* [1,3]-MISL. Given a preassociated $\langle\underline{\mathrm{H}}\rangle$ as a current input tone symbol, an underlying preassociated vowel $\langle\underline{\mathrm{V}}\rangle$ is delinked regardless of context, while current tone symbol $\langle\underline{\mathrm{H}}\rangle$ is associated with the penultimate vowel. This requires a window of size 3 on the vowel string to check if the current vowel is the penultimate vowel.

Similar to Rimi, Bemba (2h) shows bounded tone spread whereby a preassociated tone-vowel pair is not delinked but the next vowel also becomes associated to the tone: / VV́VV/ or /$\langle\underline{\mathrm{H}}\rangle$ + V$\langle\underline{\mathrm{V}}\rangle$VV/ → [VV́V́V]. This is ISL, A-ISL, and [1,2]-MISL. The only difference from Rimi is that an input preassociated vowel $\langle\underline{\mathrm{V}}\rangle$ is not delinked, i.e. it keeps its tone in the output.

In Arusa (2i), a process of unbounded deletion deletes a phrase-final H tone if it is follows another H tone. By deleting the H tone, any preassociated vowels become delinked and toneless: /V́ VV́V́V/ or /$\langle\underline{\mathrm{H}}\rangle$ $\langle\underline{\mathrm{H}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle$ V($\underline{\mathrm{VV}}$)V/ → [V́ VVVV]. This process is not ISL because of the unbounded distance between the two spans of high vowels, but it is A-ISL and [3,1]-MISL.[10] A locality window of size 3 is needed on the **T**-string in order to check if the current input tone symbol is a phrase-final $\langle\underline{\mathrm{H}}\rangle$ and succeeds another high tone. If yes, then any currently read input vowels are delinked.

4.3 Distinct functions across locality classes

The distinctions between ISL, A-ISL, and MISL are visible in more complex patterns in Table 2j-l. So far, all the A-ISL and ISL functions we described were also MISL. But some ISL yet non-A-ISL functions are *variably* MISL depending on how the function is defined. They are MISL *only* if the function generates as output two output strings of associated tones vs. associated vowels instead of only one output string (§4.3.1). Furthermore, some patterns are neither ISL, A-ISL, or MISL (§4.3.2). And finally, some patterns are MISL but neither ISL nor A-ISL (§4.3.3).

4.3.1 ISL but not A-ISL; variably MISL

Luganda (2j) has a process of bounded Meussen's rule which is ISL but not A-ISL. Here, if a preas-

[8]Note that ($\underline{\mathrm{V}}$, $\underline{\mathrm{V}}$, and $\underline{\mathrm{V}}$) are three separate input alphabet symbols.

[9]One possible system, inspired from Yli-Jyrä (2015), is to use the symbols / and \on the vowel-string. Given a tuple of [$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{L}}\rangle$, ($\underline{\mathrm{V}}$/ $\underline{\mathrm{V}}$)] where space marks the separation of multicharacter symbols, the slash / means that the first tone is associated to the first vowel while the second tone to the two vowels. Similarly for [$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{L}}\rangle$, ($\underline{\mathrm{V}}$ \$\underline{\mathrm{V}}$)], the first tone is associated with the two vowels while the second tone with the second vowel.

[10]The FST in the appendix is [3,1]-MISL but it cannot ensure that the number of preassociated tones in the input match the number of spans of preassociated vowels. Doing so requires that we either increase the locality window on the vowel tape to 2, *or* we output a string tuple such that the function changes the substring $\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle\ltimes$to $\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{L}}\rangle\ltimes$, similarly to the Luganda case in §4.3.1.

sociated H tone precedes another preassociated H tone *and* the two tones are associated to a contiguous sequence of vowels, then the second H tone becomes low: /V́V́V́V/ or /$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle(\underline{\mathrm{V}}\underline{\mathrm{V}})\mathrm{V}$/ $\rightarrow$ [V́V̀V̀V]. The function is not A-ISL because it needs to reference contiguity on both the tone and vowel strings, see Chandlee and Jardine (2019a) on why this matters.

Similarly, if the function is defined as a multi-input function which generates only *one* output string, then the function is not MISL. Assume the **T**-string is $\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle$, and the **V**-string contains two vowels preassociated to the two different tones which we *represent* with butting brackets: /$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle(\underline{\mathrm{V}}\underline{\mathrm{V}}\ldots\underline{\mathrm{V}})$/. The second vowel $(\underline{\mathrm{V}}$ will map to a surface low toned vowel V̀ because the two tones are contiguous. The second vowel $(\underline{\mathrm{V}}$ starts a span of preassociated vowels. But for the other vowels like the final $\underline{\mathrm{V}})$, an MISL function cannot keep track if this vowel was part of a preassociated vowel span which succeeded another span, i.e. it can't know if $\underline{\mathrm{V}})$ is preceded by the substring $\langle\underline{\mathrm{V}}\rangle$ $(\underline{\mathrm{V}}$ or not.

But if the function generates as output *two* output strings as an output tuple of tones and vowels, then the function is [2,2]-MISL. The input /$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle(\underline{\mathrm{V}}\ \underline{\mathrm{V}}\ \underline{\mathrm{V}})$/ is mapped to [$\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{L}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle(\underline{\mathrm{V}}\ \underline{\mathrm{V}}\ \underline{\mathrm{V}})$] with the only change being on the **T**-string. The function is [2,2]-MISL because it checks if i) the current tone symbol is a preassociated $\langle\underline{\mathrm{H}}\rangle$ and immediately succeeds another tone symbol $\langle\underline{\mathrm{H}}\rangle$ *and* if ii) the current vowel symbol is preassociated $\langle\underline{\mathrm{V}}\rangle$ or starts a span of preassociated vowels $(\underline{\mathrm{V}}$, and follows a span of preassociated vowels $\langle\underline{\mathrm{V}}\rangle$ or $\underline{\mathrm{V}})$. All this information is local with a window of 2 on the two strings.

4.3.2 Neither ISL, A-ISL, nor MISL

Shona (2k) has a process of Alternating Meussen's rule where hetero-morphemic and contiguous spans of preassociated high-toned vowels alternate to form high and low sequences: /V́-V́-V́/ $\rightarrow$ [V́-V̀-V́]. This is not ISL, A-ISL, or MISL because iterative alternation is local over output information, not input information. This is explained further in Chandlee and Jardine (2019a).

4.3.3 MISL but neither ISL nor A-ISL

Finally, Ndebele (2l) has unbounded spreading of a preassociated H tone up until the ante-penultimate vowel: /V́VVVV/ or /$\langle\underline{\mathrm{H}}\rangle$ + $\langle\underline{\mathrm{V}}\rangle$VVVV/ $\rightarrow$ [V́V́V́VV]. This process is neither ISL nor A-ISL but it is [1,3]-MISL. Reading from right-to-left, the last two vowels surface as toneless. But if the current tone symbol is a preassociated $\langle\underline{\mathrm{H}}\rangle$, then any vowel which is not the penultimate or ultimate surfaces as high V́. This requires a window of size 3 on the **V**-tape, but only 1 on the tone tape.

5 Conclusion

This paper examined the computational expressivity of autosegmental phonology, in particular tonal processes. Generalizing Input Strictly Local (ISL) functions to handle multiple inputs, we showed that the class of Multi-Input Strictly Local (MISL) functions can compute almost all attested tonal processes. These MISL functions are computed by restricted deterministic asynchronous multi-tape finite-state transducers. Using a careful linear encoding mechanism, this computational result applies equally well to tonal processes with or without preassociation. The result also narrows the gap in mathematical results between segmental and autosegmental phonology.

References

Steven Bird. 1995. Computational phonology: a constraint-based approach. Studies in Natural Language Processing. Cambridge University Press, Cambridge.

Steven Bird and T Mark Ellison. 1994. One-level phonology: Autosegmental representations and rules as finite automata. Computational Linguistics, 20(1):55–90.

Steven Bird and Ewan Klein. 1990. Phonological events. Journal of linguistics, 26(1):33–56.

Jane Chandlee. 2014. Strictly Local Phonological Processes. Ph.D. thesis, University of Delaware, Newark, DE.

Jane Chandlee. 2017. Computational locality in morphological maps. Morphology, pages 1–43.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015a. Output strictly local functions. In 14th Meeting on the Mathematics of Language, pages 112–125.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015b. Output strictly local functions. In Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), pages 112–125, Chicago, USA.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. Linguistic Inquiry, 49(1):23–60.

Jane Chandlee and Adam Jardine. 2019a. Autosegmental input strictly local functions. Transactions of the Association for Computational Linguistics, 7:157–168.

Jane Chandlee and Adam Jardine. 2019b. Quantifier-free least fixed point functions for phonology. In Proceedings of the 16th Meeting on the Mathematics of Language (MoL 16), Toronto, Canada. Association for Computational Linguistics.

John Coleman. 1998. Phonological representations: their names, forms and powers. Cambridge University Press.

John Coleman and John Local. 1991. The no crossing constraint in autosegmental phonology. Linguistics and Philosophy, 14(3):295–338.

C. C. Elgot and J. E. Mezei. 1965. On relations defined by generalized finite automata. IBM Journal of Research and Development, 9(1):47–68.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. ACM SIGLOG News, 3(3):4–19.

Patrick C Fischer. 1965. Multi-tape and infinite-state automataa survey. Communications of the ACM, 8(12):799–805.

Patrick C Fischer and Arnold L Rosenberg. 1968. Multitape one-way nonwriting automata. Journal of Computer and System Sciences, 2(1):88–101.

Christiane Frougny and Jacques Sakarovitch. 1993. Synchronized rational relations of finite and infinite words. Theoretical Computer Science, 108(1):45–82.

Carlo A. Furia. 2012. A survey of multi-tape automata. http://arxiv.org/abs/1205.0178. Latest revision: November 2013.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13), pages 52–63, Sofia, Bulgaria. Association for Computational Linguistics.

Adam Jardine. 2016a. Computationally, tone is different. Phonology, 33(2):247–283.

Adam Jardine. 2016b. Locality and non-linear representations in tonal phonology. Ph.D. thesis, University of Delaware, Newark, DE.

Adam Jardine. 2017a. The local nature of tone-association patterns. Phonology, 34(2):363–384.

Adam Jardine. 2017b. On the logical complexity of autosegmental representations. In Proceedings of the 15th Meeting on the Mathematics of Language, pages 22–35.

Adam Jardine. 2019. The expressivity of autosegmental grammars. Journal of Logic, Language and Information, 28(1):9–54.

Martin Kay. 1987. Nonconcatenative finite-state morphology. In Third Conference of the European Chapter of the Association for Computational Linguistics.

George Anton Kiraz. 2001. Computational nonlinear morphology: with emphasis on Semitic languages. Cambridge University Press.

Andras Kornai. 1995. Formal phonology. Garland Publishing Inc.

Nathan Koser, Christopher Oakden, and Adam Jardine. 2019. Tone association and output locality in non-linear structures. In Supplemental proceedings of AMP 2019.

Robert McNaughton and Seymour A Papert. 1971. Counter-Free Automata (MIT research monograph no. 65). The MIT Press.

Michael O Rabin and Dana Scott. 1959. Finite automata and their decision problems. IBM journal of research and development, 3(2):114–125.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In Formal Grammar, volume 8036 of Lecture Notes in Computer Science, pages 90–108. Springer.

James Rogers and Geoffrey Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. Journal of Logic, Language and Information, 20:329–342.

Bruce Wiebe. 1992. Modelling autosegmental phonology with multi-tape finite state transducers. Master's thesis, Simon Fraser University.

Anssi Yli-Jyrä. 2013. On finite-state tonology with autosegmental representations. In Proceedings of the 11th international conference on finite state methods and natural language processing. Association for Computational Linguistics.

Anssi Yli-Jyrä. 2015. Three equivalent codes for autosegmental representations. In Proceedings of the 12th International Conference on Finite-State Methods and Natural Language Processing 2015 (FSMNLP 2015 Düsseldorf).

A Appendix

A sample MT-FST and derivation are given for some of the tone processes.

A.1 Tonal processes without preassociation

These patterns take as input a pair of strings without preassociation.

A.1.1 Kikuyu spreading

In Kikuyu (Table 2b), the first tone associates with the first two vowels. 1-to-1 association follows. A final tone may undergo final spreading, e.g. f([LHLH, VVVVVVV])=$\grave{V}\grave{V}\acute{V}\grave{V}\acute{V}\acute{V}\acute{V}$. A [2,3]-MISL MT-FST is provided in Figure 3, with a sample derivation in Table 4.

A.1.2 Hausa right-to-left spreading

In Hausa (Table 2b), tones are associated right-to-left with initial spread, e.g. f([LH, VVV])=$\grave{V}\grave{V}\acute{V}$. This function is modeled by the [2,1]-MISL MT-FST in Figure 4, with a sample derivation in Table 4. The FST processes the input string-tuple from right to left using the -1 direction parameter.

A.2 Tonal processes with preassociation

These functions take as input a preassociated pair of tones and vowels.

A.2.1 Rimi bounded tone shift

In Rimi (Table 2f), a preassociated tone will shift one vowel to the right, e.g. f([$\langle\underline{\text{H}}\rangle$, V$\langle\underline{\text{V}}\rangle$VV]=$VV\acute{V}V$. This function is modeled by the [1,2]-MISL MT-FST in Figure 5, with a sample derivation in Table 6. We assume that the only possible underlying tone string is a preassociated H.

Final preassociated vowels do not undergo tone shift: f([$\langle\underline{\text{H}}\rangle$, VVV$\langle\underline{\text{V}}\rangle$]=$VVV\acute{V}$. We factor this out for illustrative reasons. Otherwise, the function is [2,2]-MISL and needs a MT-FST with more states.

A.2.2 Zigulu unbounded tone shift

In Zigulu (Table 2g), unbounded tone shift causes a preassociated H tone to shift to the penultimate vowel, e.g. f([$\langle\underline{\text{H}}\rangle$, VV$\langle\underline{\text{V}}\rangle$VVV])=$VVVV\acute{V}V$. This function is modeled by the [1,3]-MISL MT-FST in Figure 6, with a sample derivation in 7. For easier illustration, the MT-FST processes the input right-to-left using the -1 direction parameter. We assume that the tone string can either be an empty string $\rtimes\lambda\ltimes$ or a single preassociated H tone $\rtimes\langle\underline{H}\rangle\ltimes$.

A.2.3 Bemba unbounded tone spread

In Bemba (Table 2h), bounded tone spread causes a preassociated H tone to surface on its preassociated vowel and on the subsequent vowel, e.g. f([$\langle\underline{\text{H}}\rangle$,V$\langle\underline{\text{V}}\rangle$VV])=$V\acute{V}\acute{V}V$. This function is modeled by the [1,2]-MISL MT-FST in Figure 7, with a sample derivation in Table 8. We assume that the input tone string contains either an empty string $\rtimes\lambda\ltimes$ or a single preassociated H tone $\rtimes\langle\underline{H}\rangle\ltimes$.

A.2.4 Arusa unbounded deletion

In Aursa (Table 2i), unbounded deletion causes a phrase-final preassociated H to delete if it follows another H tone, e.g. f([$\langle\underline{\text{H}}\rangle\langle\underline{\text{H}}\rangle$,$\langle\underline{\text{V}}$ V($\underline{\text{VV}}$V]=$\acute{V}$ VVVV. This function is computed by the [3,1]-MISL MT-FST in Figure 8, with a sample derivation in 9. The FST reads the input from right-to-left using the -1 direction parameter. We assume the input tone string contains zero or more preassociated H tones: **T**=$\rtimes\langle\underline{\text{H}}\rangle^*\ltimes$.

As a caveat, the function in Figure () cannot ensure that the number of preassociated tones matches the number of spans of preassociated vowels. That more faithful function is [3,2]-MISL. We do not draw it here because of size.

For clarity, in Table 9, preassociated vowels are given a subscript $_1$ instead of underlining.

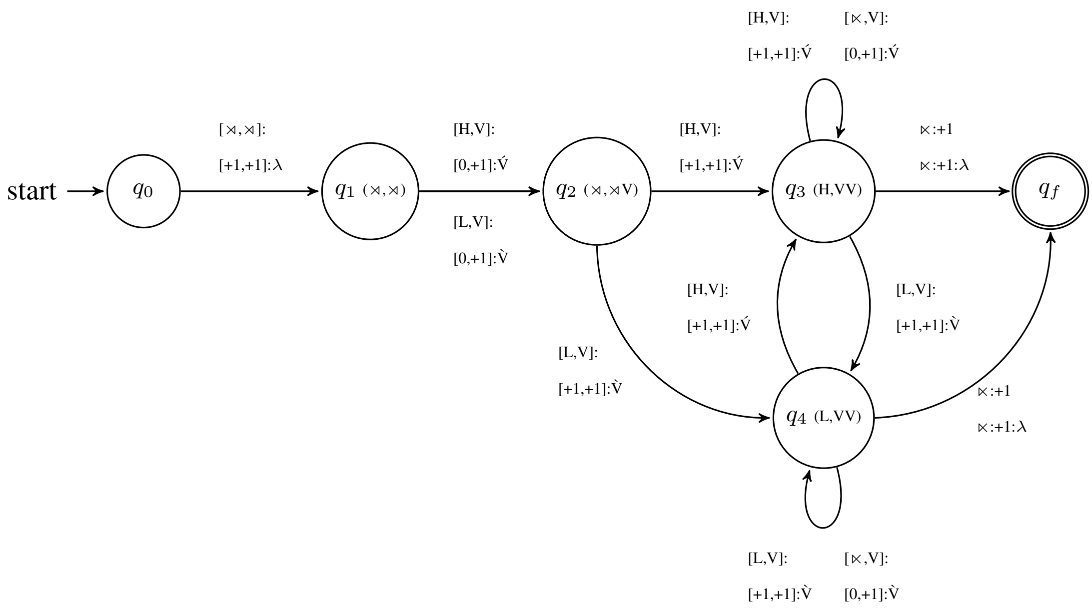


Figure 3: MT-FST for Kikuyu initial spread

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\underline{\rtimes}LHLH\ltimes$	$\underline{\rtimes}VVVVVVV\ltimes$		
2.	q_1	$\rtimes\underline{L}HLH\ltimes$ $\rtimes$:+1	$\rtimes\underline{V}VVVVVV\ltimes$ $\rtimes$:+1	λ	
3.	q_2	$\rtimes\underline{L}HLH\ltimes$ L:0	$\rtimes V\underline{V}VVVVV\ltimes$ V:+1	V̀	V̀
4.	q_4	$\rtimes L\underline{H}LH\ltimes$ L:+1	$\rtimes VV\underline{V}VVVV\ltimes$ V:+1	V̀	V̀V̀
5.	q_3	$\rtimes LH\underline{L}H\ltimes$ H:+1	$\rtimes VVV\underline{V}VVV\ltimes$ V:+1	V́	V̀V̀V́
6.	q_4	$\rtimes LHL\underline{H}\ltimes$ L:+1	$\rtimes VVVV\underline{V}VV\ltimes$ V:+1	V̀	V̀V̀V́V̀
7.	q_3	$\rtimes LHLH\underline{\ltimes}$ H:+1	$\rtimes VVVVV\underline{V}V\ltimes$ V:+1	V́	V̀V̀V́V̀V́
8.	q_3	$\rtimes LHLH\underline{\ltimes}$ $\ltimes$:0	$\rtimes VVVVVV\underline{V}\ltimes$ V:+1	V́	V̀V̀V́V̀V́V́
9.	q_3	$\rtimes LHLH\underline{\ltimes}$ $\ltimes$:0	$\rtimes VVVVVVV\underline{\ltimes}$ V:+1	V́	V̀V̀V́V̀V́V́V́
10.	q_f	$\rtimes LHLH\ltimes$ $\ltimes$:+1	$\rtimes VVVVVVV\ltimes$ $\ltimes$:+1	λ	V̀V̀V́V̀V́V́V́

Table 4: Derivation of f([LHLH, VVVVVVV])=*V̀V̀V́V̀V́V́V́* in Kikuyu with the MT-FST in Figure 3

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\rtimes LH\underline{\ltimes}$	$\rtimes VVV\underline{\ltimes}$		
2.	q_1	$\rtimes L\underline{H}\ltimes$ $\ltimes$:-1	$\rtimes VV\underline{V}\ltimes$ $\ltimes$:-1	λ	
3.	q_2	$\rtimes\underline{L}H\ltimes$ H:-1	$\rtimes V\underline{V}V\ltimes$ V:-1	V́	V́
4.	q_3	$\underline{\rtimes}LH\ltimes$ L:-1	$\rtimes\underline{V}VV\ltimes$ V:-1	V̀	V̀V́
5.	q_3	$\underline{\rtimes}LH\ltimes$ $\rtimes$:0	$\underline{\rtimes}VVV\ltimes$ V:-1	V̀	V̀V̀V́
6.	q_f	$\rtimes LH\ltimes$ $\rtimes$:-1	$\rtimes VVV\ltimes$ $\rtimes$:-1	λ	V̀V̀V́

Table 5: Derivation of f([LH, VVV])=*V̀V̀V́* in Hausa with the MT-FST in Figure 4

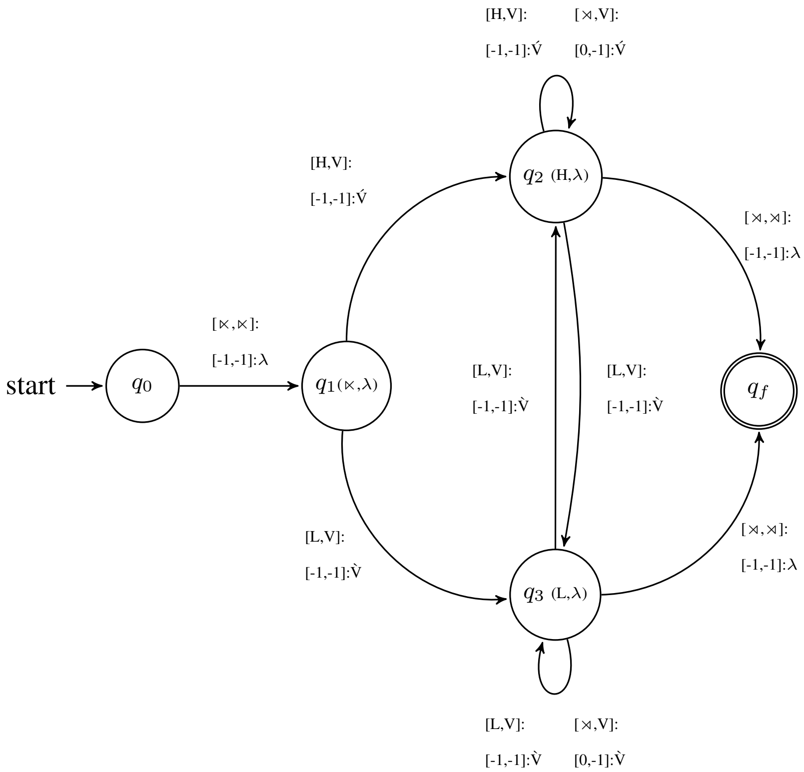


Figure 4: MT-FST for Hausa

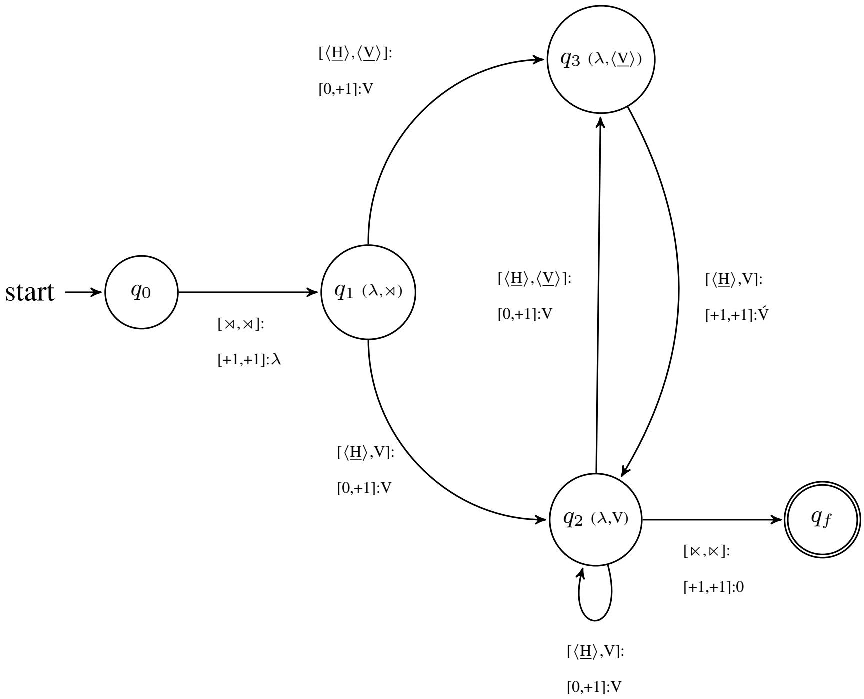


Figure 5: MT-FST for Rimi

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$	$\underline{\rtimes}\text{V}\langle\text{V}\rangle\text{VV}\ltimes$		
2.	q_1	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\rtimes$:+1	$\rtimes\underline{\text{V}}\langle\text{V}\rangle\text{VV}\ltimes$ $\quad\rtimes$:+1	λ	
3.	q_2	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\langle H\rangle$:0	$\rtimes\text{V}\underline{\langle\text{V}\rangle}\text{VV}\ltimes$ $\quad V$:+1	V	V
4.	q_3	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\langle H\rangle$:0	$\rtimes\text{V}\langle\text{V}\rangle\underline{\text{V}}\text{V}\ltimes$ $\quad\langle V\rangle$:+1	V	VV
5.	q_2	$\rtimes\langle\text{H}\rangle\underline{\ltimes}$ $\quad\langle H\rangle$:+1	$\rtimes\text{V}\langle\text{V}\rangle\text{V}\underline{\text{V}}\ltimes$ $\quad V$:+1	V́	VVV́
6.	q_2	$\rtimes\langle\text{H}\rangle\underline{\ltimes}$ $\quad\ltimes$:0	$\rtimes\text{V}\langle\text{V}\rangle\text{VV}\underline{\ltimes}$ $\quad V$:+1	V	VVV́V
7.	q_f	$\rtimes\langle\text{H}\rangle\ltimes$ $\quad\ltimes$:+1	$\rtimes\text{V}\langle\text{V}\rangle\text{VV}\ltimes$ $\quad\ltimes$:+1	λ	VVV́V

Table 6: Derivation of $f([\langle\underline{\text{H}}\rangle, \text{V}\langle\underline{\text{V}}\rangle\text{VV}]$=*VVV́V* in Rimi with the MT-FST in Figure 5

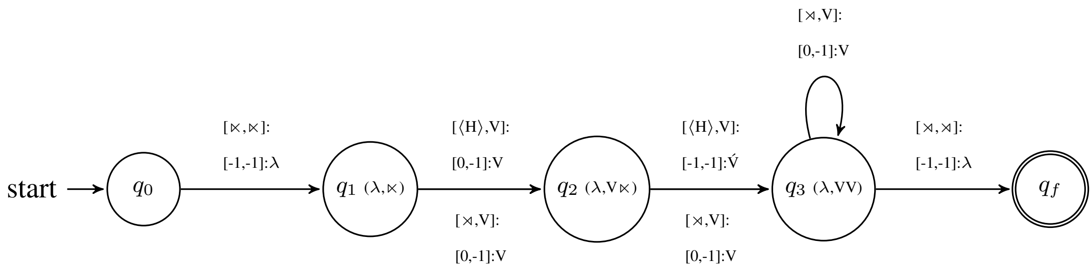


Figure 6: MT-FST for Zigulu

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\rtimes\langle\text{H}\rangle\underline{\ltimes}$	$\rtimes\text{VVVVVV}\underline{\ltimes}$		
2.	q_1	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\ltimes$:-1	$\rtimes\text{VVVVV}\underline{\text{V}}\ltimes$ $\quad\ltimes$:-1	λ	
3.	q_2	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\langle H\rangle$:0	$\rtimes\text{VVVV}\underline{\text{V}}\text{V}\ltimes$ $\quad V$:-1	V	V
4.	q_3	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$ $\quad\langle H\rangle$:-1	$\rtimes\text{VVV}\underline{\text{V}}\text{VV}\ltimes$ $\quad V$:-1	V́	V́V
5.	q_3	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$ $\quad\rtimes$:0	$\rtimes\text{VV}\underline{\text{V}}\text{VVV}\ltimes$ $\quad V$:-1	V	VV́V
6.	q_3	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$ $\quad\rtimes$:0	$\rtimes\text{V}\underline{\text{V}}\text{VVVV}\ltimes$ $\quad V$:-1	V	VVV́V
7.	q_3	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$ $\quad\rtimes$:0	$\rtimes\underline{\text{V}}\text{VVVVV}\ltimes$ $\quad V$:-1	V	VVVV́V
8.	q_3	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$ $\quad\rtimes$:0	$\underline{\rtimes}\text{VVVVVV}\ltimes$ $\quad V$:-1	V	VVVVV́V
9.	q_f	$\rtimes\langle\text{H}\rangle\ltimes$ $\quad\rtimes$:-1	$\rtimes\text{VVVVVV}\ltimes$ $\quad\rtimes$:-1	λ	VVVVV́V

Table 7: Derivation of $f([\langle\underline{\text{H}}\rangle, \text{VV}\langle\underline{\text{V}}\rangle\text{VVV}])$=*VVVVV́V* in Zigulu with the MT-FST in Figure 6

	Current state	Tone tape	Vowel tape	Output symbol	Output string
1.	q_0	$\underline{\rtimes}\langle\text{H}\rangle\ltimes$	$\underline{\rtimes}\text{V}\langle\text{V}\rangle\text{VV}\ltimes$		
2.	q_1	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\rtimes$:+1	$\rtimes\underline{\text{V}}\langle\text{V}\rangle\text{VV}\ltimes$ $\quad\rtimes$:+1	λ	
3.	q_2	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\langle H\rangle$:0	$\rtimes\text{V}\underline{\langle\text{V}\rangle}\text{VV}\ltimes$ $\quad V$:+1	V	V
4.	q_3	$\rtimes\underline{\langle\text{H}\rangle}\ltimes$ $\quad\langle H\rangle$:0	$\rtimes\text{V}\langle\text{V}\rangle\underline{\text{V}}\text{V}\ltimes$ $\quad\langle V\rangle$:+1	V	VV́
5.	q_2	$\rtimes\langle\text{H}\rangle\underline{\ltimes}$ $\quad\langle H\rangle$:+1	$\rtimes\text{V}\langle\text{V}\rangle\text{V}\underline{\text{V}}\ltimes$ $\quad V$:+1	V́	VV́V́
6.	q_2	$\rtimes\langle\text{H}\rangle\underline{\ltimes}$ $\quad\ltimes$:0	$\rtimes\text{V}\langle\text{V}\rangle\text{VV}\underline{\ltimes}$ $\quad V$:+1	V	VV́V́V
7.	q_f	$\rtimes\langle\text{H}\rangle\ltimes$ $\quad\ltimes$:+1	$\rtimes\text{V}\langle\text{V}\rangle\text{VV}\ltimes$ $\quad\ltimes$:+1	λ	VV́V́V

Table 8: Derivation of $f([\langle\underline{\text{H}}\rangle, \text{V}\langle\underline{\text{V}}\rangle\text{VV}]$=*VV́V́V* in Bemba with the MT-FST in Figure 7

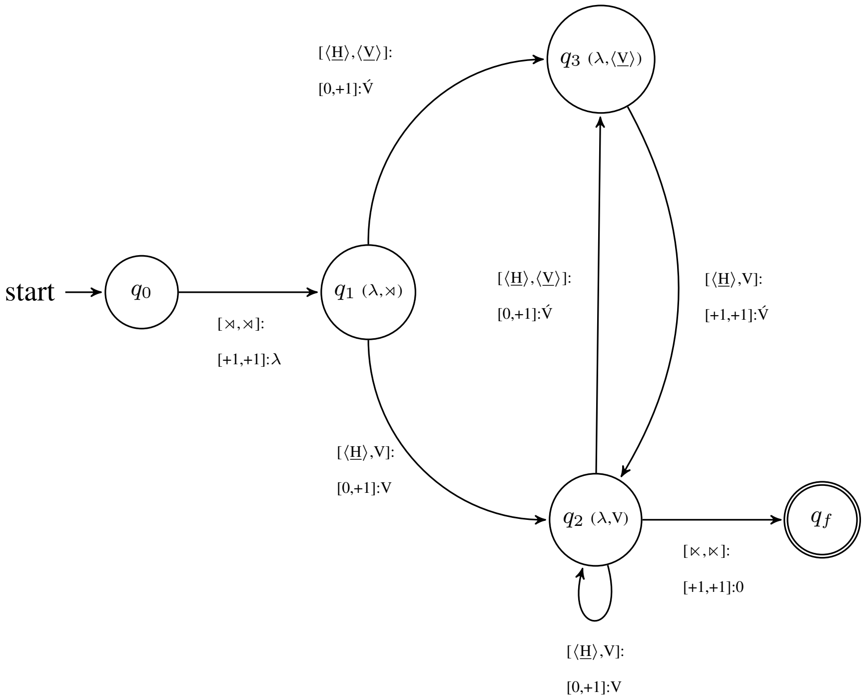


Figure 7: MT-FST for Bemba

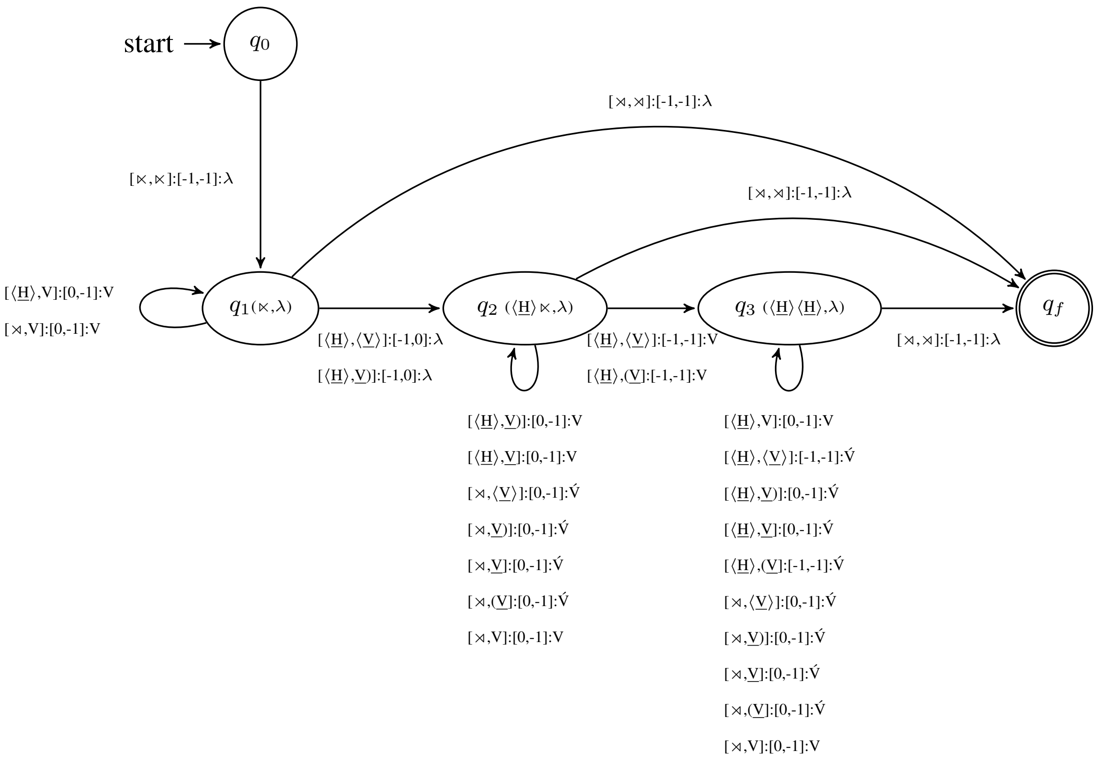


Figure 8: MT-FST for Arusa

	Current state	Tone tape		Vowel tape		Output symbol	Output string
1.	q_0	$\rtimes\langle \mathrm{H}\rangle\langle \mathrm{H}\rangle\underline{\ltimes}$		$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\mathrm{V}_1)\mathrm{V}\underline{\ltimes}$			
2.	q_1	$\rtimes\langle \mathrm{H}\rangle\underline{\langle \mathrm{H}\rangle}\ltimes$	$\ltimes$:-1	$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\mathrm{V}_1)\underline{\mathrm{V}}\ltimes$	$\ltimes$:-1	λ	
3.	q_1	$\rtimes\langle \mathrm{H}\rangle\underline{\langle \mathrm{H}\rangle}\ltimes$	$\langle H\rangle$:0	$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\underline{\mathrm{V}_1})\mathrm{V}\ltimes$	V:-1	V	V
4.	q_2	$\rtimes\underline{\langle \mathrm{H}\rangle}\langle \mathrm{H}\rangle\ltimes$	$\langle H\rangle$:-1	$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\underline{\mathrm{V}_1)}\mathrm{V}\ltimes$	$V_1)$:-1	λ	V
5.	q_2	$\rtimes\underline{\langle \mathrm{H}\rangle}\langle \mathrm{H}\rangle\ltimes$	$\langle H\rangle$:0	$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}\underline{(\mathrm{V}_1}\mathrm{V}_1)\mathrm{V}\ltimes$	$V_1)$:-1	V	VV
6.	q_3	$\underline{\rtimes}\langle \mathrm{H}\rangle\langle \mathrm{H}\rangle\ltimes$	$\langle H\rangle$:-1	$\rtimes\langle \mathrm{V}_1\rangle \underline{\mathrm{V}}(\mathrm{V}_1\mathrm{V}_1)\mathrm{V}\ltimes$	$(V_1$:-1	V	VVV
7.	q_3	$\underline{\rtimes}\langle \mathrm{H}\rangle\langle \mathrm{H}\rangle\ltimes$	$\rtimes$:0	$\rtimes\underline{\langle \mathrm{V}_1\rangle} \mathrm{V}(\mathrm{V}_1\mathrm{V}_1)\mathrm{V}\ltimes$	$(V$:-1	V	VVVV
8.	q_3	$\underline{\rtimes}\langle \mathrm{H}\rangle\langle \mathrm{H}\rangle\ltimes$	$\rtimes$:0	$\underline{\rtimes}\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\mathrm{V}_1)\mathrm{V}\ltimes$	$\langle V_1\rangle$:-1	V́	V́VVVV
9.	q_f	$\rtimes\langle \mathrm{H}\rangle\langle \mathrm{H}\rangle\ltimes$	$\rtimes$:-1	$\rtimes\langle \mathrm{V}_1\rangle \mathrm{V}(\mathrm{V}_1\mathrm{V}_1)\mathrm{V}\ltimes$	$\rtimes$:-1	λ	V́VVVV

Table 9: Derivation of $f([\langle\underline{\mathrm{H}}\rangle\langle\underline{\mathrm{H}}\rangle,\langle\underline{\mathrm{V}}\ \mathrm{V}(\underline{\mathrm{VV}}\mathrm{V}]=\acute{V}\ VVVV$ in Arusa with the MT-FST in Figure 8

Metrical Grids and Generalized Tier Projection

Yiding Hao
Yale University
New Haven, CT, USA
yiding.hao@yale.edu

Abstract

This paper formalizes metrical grid theory (MGT, Prince, 1983; Hayes, 1995) and studies its expressive power. I show that MGT analyses of a certain form can describe stress systems beyond the *input tier-based input strictly local* functions proposed by Hao and Andersson (2019), but conjecture that such analyses do not describe systems beyond the *input tier-based strictly local* languages of Baek (2018). These results reveal fundamental differences between the three formalisms.

1 Introduction

The problem of unbounded stress has proven challenging for subregular phonology. Baek (2018) has recently shown that unbounded stress provides a counterexample to the *weak subregular hypothesis* (Heinz, 2018), which claims that phonological phenomena can be represented by *tier-based strictly local languages* (TSL, Heinz et al., 2011) when viewed as *decision problems* that accept grammatical utterances. To remedy this, Baek proposes that the decision to project a symbol to the tier may be conditioned by local contextual information such as the presence of word boundaries. This proposal has been extended to a *generalized tier projection* system in which tier projection is implemented by arbitrary deleting functions (Mayer and Major, 2018; Graf and Mayer, 2018).

Viewing unbounded stress as a *transduction problem* that maps underlying forms without prosodic representation to surface forms marked for primary stress, Hao and Andersson (2019) show that unbounded stress systems are similarly handeled by generalized tier projection, but that they fall outside the class of *tier-based input strictly local functions* (TISL, Chandlee, 2014). Hao and Andersson demonstrate that *default-to-opposite-side* (DO) systems can be captured by adapting generalized tier projection to subregular functions. However, it turns out that the "bidirectional" nature of *default-to-same-side* (DS) systems cannot be implemented by subsequential functions, which allow only a single unidirectional pass over the input. They instead propose that DS systems should be understood as *interaction-free weakly deterministic functions* in the sense of McCollum et al. (2018).

These results raise conceptual questions regarding the treatment of stress in subregular phonology. In particular, the mechanism of generalized tier projection intuitively seems powerful and *ad-hoc*. The basic elements of subregular phonology, namely strict locality and the traditional tier projection system of Heinz et al. (2011), can be viewed as formalizations of rule and tier systems that are well-established in phonological theory (Chandlee, 2014; Chomsky and Halle, 1968; Goldsmith, 1976). While generalized tier projection enables Baek (2018) and Hao and Andersson (2019) to define classes of finite-state machines that capture unbounded stress, it is unclear whether generalized tier projection is similarly grounded in existing phonological constructs. Reflecting on this issue, Hao and Andersson observe that the generalized tier-projection mechanism they use to produce *Dybo's Rule* (Dybo, 1977), a model of the stress system in Abkhaz, bears a striking resemblance to the syllable tier used in a standard analysis of that system within *metrical grid theory* (MGT, Prince, 1983; Hayes, 1995). From that observation we might hypothesize that stress systems computed using generalized tier projection naturally correspond to those described by MGT.

In this paper, I will argue that this intuition does not hold for the transduction problem, though it may hold for the decision problem. To that end, I define a formal model of MGT in Section 4, and show in Section 5 that the stress systems described by MGT do not correspond to those represented by functions defined using generalized tier projection. I also give evidence to suggest that decision problems described by MGT can be represented using generalized tier projection even when their corresponding transduction problems cannot. These results imply not only that functions based on generalized tier projection are not grounded in MGT, but also that the typological predictions they make about the range of possible stress systems differ from those made by MGT and by decision problems.

Technical definitions used in this paper are given in Section 2, and Section 3 reviews the existing results on unbounded stress in subregular phonology. Section 6 concludes.

2 Preliminaries

In this paper, uppercase Greek letters denote finite alphabets not including the boundary symbols $\rtimes$ and $\ltimes$. The length of a string x is denoted by $|x|$, and λ denotes the empty string. Alphabet symbols are identified with strings of length 1, and individual strings are identified with singleton sets of strings. For $k \geq 0$, α^k denotes α concatenated with itself k-many times, $\alpha^{<k}$ denotes $\bigcup_{i=0}^{k-1} \alpha^i$, α^* denotes $\bigcup_{i=0}^{\infty} \alpha^i$, and α^+ denotes $\alpha\alpha^*$. The *longest common prefix* of a set of strings A is the longest string $\mathrm{lcp}(A)$ such that every string in A begins with $\mathrm{lcp}(A)$.

For sets A and B, the notation $f : A \to B$ means that f is a function with domain A and codomain B. The *range of* f is the set $\{y|\exists x.f(x) = y\} \subseteq B$. A function $f : A \to B$ is *injective* if for every $x, y \in A$, $f(x) = f(y)$ if and only if $x = y$. A function $f : \Sigma^* \to \Gamma^*$ is *same-length* if and only if for all $x \in \Sigma^*$, $|f(x)| = |x|$.

A *subsequential finite-state transducer* (SFST) is a 6-tuple $T = \langle Q, \Sigma, \Gamma, q_0, \to, \omega\rangle$, where

- Q is the set of *states*, with $q_0 \in Q$ being the *start state*;
- Σ and Γ are the *input* and *output alphabets*, respectively;
- $\to : Q \times \Sigma \to Q \times \Gamma^*$ is the *transition function*; and
- $\omega : Q \to \Gamma^*$ is the *final output function*.

For $x \in \Sigma^*$; $y \in \Gamma^*$; and $q, r \in Q$, the notation $q \xrightarrow{x:y} r$ means that T emits y to the output stream and transitions to state r if it reads x in the input stream while it is in state q. Letting $f : \Sigma^* \to \Gamma^*$, we say that T *computes* f if for every $x \in \Sigma^*$, $f(x) = y\omega(q)$, where $q_0 \xrightarrow{x:y} q$. A function is *subsequential* if it is computed by an SFST.

For a string $x \neq \lambda$, I use the following indexing notation.

- For $1 \leq i \leq j \leq |x|$, $x[i : j]$ is the substring of x such that $x = wx[i : j]y$, where $|w| = i - 1$ and $|y| = |x| - j$.
- For $-|x| \leq u, v \leq |x|$ and $1 \leq i \leq j \leq |x|$, $x[u : v] = x[i : j]$ if $u \equiv i \mod (|x| + 1)$ and $v \equiv j \mod (|x| + 1)$.
- For each i, $x[i] := x[i : i]$; $x[i :] := x[i : |x|]$; and $x[: i] := x[1 : i]$.

The remainder of this section reviews the algebraic characterization of subsequential functions as well as tier projection and strict locality.

2.1 Subsequential Functions

Independently of SFSTs, the subsequential functions can be characterized using two operations on string functions.

Definition 1. Let $f : \Sigma^* \to \Gamma^*$. We define the function $f^{\leftarrow} : \Sigma^* \to \Gamma^*$ by

$$f^{\leftarrow}(x) := \mathrm{lcp}\left(\{f(xy)|y \in \Sigma^*\}\right).$$

For any $x, y \in \Sigma^*$, $f_x^{\to}(y)$ denotes the string such that $f(xy) = f^{\leftarrow}(x)f_x^{\to}(y)$. We refer to $f_x^{\to}$ as the *translation of* f *by* x and to $f^{\leftarrow}$ as f *top*.

The translations of a subsequential function may be used to construct the minimal SFST for that function, analogously to the Nerode–Myhill construction for the minimal finite-state automaton of a regular language.

Theorem 2 (Raney, 1958)**.** *A function $f : \Sigma^* \to \Gamma^*$ is subsequential if and only if the set $\{f_x^{\to}|x \in \Sigma^*\}$ is finite.*

For a subsequential function f with minimal SFST T, the translations of f are in bijection with the states of T. After reading input x, T outputs $f^{\leftarrow}(x)$ and enters the state corresponding to $f_x^{\to}$.

2.2 Homomorphisms

This paper will frequently make use of a class of functions known as *homomorphisms*.

Definition 3. A function $h : \Sigma^* \to \Gamma^*$ is a *homomorphism* if for every $x, y \in \Sigma^*$, $h(xy) = h(x)h(y)$.

Intuitively, homomorphisms are functions that replace each symbol of Σ with a string in Γ^*. As such, homomorphisms are completely determined by their values on the input alphabet.

Proposition 4. *Let $h, g : \Sigma^* \to \Gamma^*$ be homomorphisms. If $h(x) = g(x)$ for each $x \in \Sigma$, then $h = g$.*

2.3 Locality and Tier Projection

Tier projections are functions that delete certain symbols in an input string x. A tier τ can be used to enhance notions of locality defined by grammars, automata, and transducers by having local dependencies be enforced between adjacent symbols in $\tau(x)$ instead of x, effectively ignoring symbols deleted by τ.

Definition 5. A *tier projection* is a function $\tau : \Sigma^* \to \Sigma^*$ such that $\tau(\lambda) = \lambda$ and for all $x \in \Sigma^+$, $\tau(x) = y_1 y_2 \dots y_{|x|}$, where for each i, y_i is either $x[i]$ or λ. If τ is a homomorphism, then we identify τ with the subset $\Delta \subseteq \Sigma$ such that for all $\delta \in \Delta$, $\tau(\delta) = \delta$.

Symbols not deleted by a tier projection are said to be *projected to the tier*. Tier-based strictly local functions are defined to be functions computed by minimal SFSTs whose states record the most recent $k - 1$ symbols projected to some tier, for some $k > 0$. In this paper, we assume that the states only record symbols from the SFST input projected to the tier; variants of the definitions below where the tier projects symbols of the output have also been defined (Chandlee, 2014; Chandlee et al., 2015; Burness and McMullin, 2019).

Definition 6. Let $k > 0$, and let $\tau : \Sigma^* \to \Sigma^*$ be a tier projection. A function $f : \Sigma^* \to \Gamma^*$ is *generalized input strictly k-local on tier τ* (k-GTISL on tier τ) if for all $x, y \in \Sigma^*$,

$$\tau^{\leftarrow}(x)[: 1 - k] = \tau^{\leftarrow}(y)[: 1 - k]$$

implies $f_x^{\rightarrow} = f_y^{\rightarrow}$. We say that f is

- *input strictly k-local* (k-ISL) if τ is the identity function;[1]
- *input strictly k-local on tier τ* (k-TISL on tier τ) if τ is a homomorphism; and
- *j-input strictly k-local on tier τ* (j-I-k-TISL on tier τ) if τ is j-TISL.

Remark 7. Homomorphisms are 1-ISL functions.

Tier-based strictly local languages are defined to be sets of strings whose images under some tier projection only contain substrings deemed permissible.

Definition 8. Let $k > 0$, and let $\tau : \Sigma^* \to \Sigma^*$ be a tier projection. A language $L \subseteq \Sigma^*$ is *generalized strictly k-local on tier τ* (k-GTSL on tier τ) if there exists $S \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ such that for all $x \in \Sigma^*$, $x \in L$ if and only if every length-k substring of $\rtimes^{k-1}\tau(x)\ltimes^{k-1}$ is in S. We say that L is

- *strictly k-local* (k-SL) if τ is the identity function;
- *strictly k-local on tier τ* (k-TSL on tier τ) if τ is a homomorphism; and
- *j-input strictly k-local on tier τ* (j-I-k-TSL on tier τ) if τ is j-ISL.

3 Stress in Subregular Phonology

Stress is a phonological process in which syllables are assigned varying levels of prominence (i.e., primary stress, secondary stress, or no stress) with respect to one another. Stress is *culminative*, meaning that each word contains exactly one maximally-prominent syllable. Stress is usually represented by marking syllables with their prominence levels, leaving all other information about those syllables intact. This section introduces the formalism I use to represent stress and reviews the current results on stress in subregular phonology.

Throughout this paper, I treat syllables as atomic units, and I represent them using symbols drawn from an alphabet Σ. Words, being strings of syllables, are elements of Σ^*. When a syllable $\sigma \in \Sigma$ is assigned primary stress, I mark this syllable with a diacritic $\acute{\sigma}$. I do not mark syllables for secondary stress. Thus, the set $\acute{\Sigma} := (\Sigma \cup \{\acute{\sigma} | \sigma \in \Sigma\})^*$ is the complete alphabet of symbols used to discuss stress.

Using this representation, stress in a particular language can be construed in two ways. Firstly, we may think of a stress system as a function mapping words without stress marking to words with stress

[1] In the automata theory literature, k-ISL functions are known as *k-local functions* (Vaysse, 1986). See Sakarovitch (2009, pp. 661–664) for an overview.

marking. This formalizes the transduction problem for stress.

Definition 9. A *stress system* is a same-length function $s : \Sigma^* \to \acute{\Sigma}^*$ such that for every $x \in \Sigma^+$, there exists $i > 0$ and $\sigma \in \Sigma$ such that

- $x[i] = \sigma$ and $s(x)[i] = \acute{\sigma}$; and
- for all $j > 0$, if $j \neq i$, then $s(x)[j] = x[j]$.

Remark 10. All stress systems are injective.

Secondly, we can think of a stress system as the set of all words in which stress has been assigned correctly. This formalizes the decision problem for stress.

Definition 11. A *stress constraint* is any subset $C \subseteq \acute{\Sigma}$ such that C is the range of some stress system $s : \Sigma^* \to \acute{\Sigma}^*$.

These formalizations are "equivalent" in the sense that we can easily convert between them.

Definition 12. Let $s : \Sigma^* \to \acute{\Sigma}^*$ be a stress system. The *stress constraint given by* s is the range of s.

Definition 13. Let $C \subseteq \acute{\Sigma}^*$ be a stress constraint. The *stress system given by* C is the stress system $s_C : \Sigma^* \to \acute{\Sigma}^*$ whose range is C.

A well-known example of a stress system is the *leftmost heavy otherwise rightmost* (LHOR) system (Hayes, 1995). In this system, syllables are either *light* or *heavy*. The leftmost heavy syllable in a word receives primary stress. If there are no heavy syllables in a word, then the rightmost (light) syllable receives primary stress. In Kwak'wala, for example, heavy syllables are those that contain a long vowel or a vowel with a coda consisting of [m], [n], or [l] (Bach, 1975). Some illustrative examples are given below.

(14) LHOR stress in Kwak'wala (Boas et al., 1947; Bach, 1975)

a. [tsˀəmaˈaːtud] 'to melt away something in the ear'

b. [ˈbaːbagwəm] 'boys'

c. [gəgəˈnəm] 'wives'

LHOR stress can be formalized using the following stress system. Heavy syllables are represented by the symbol H, while light syllables are represented by the symbol L.

Definition 15. Let $\Sigma := \{\mathrm{H}, \mathrm{L}\}$. The *LHOR system* is defined as follows. For $u \in \mathrm{L}^*$, $v \in \Sigma$, and $w \in \Sigma^*$,

$$\mathrm{LHOR}(uvw) = \begin{cases} u\acute{v}w, & v = \mathrm{H} \\ uw\acute{v}, & uvw \in \mathrm{L}^+. \end{cases}$$

It is easy to see that the transduction problem for LHOR is I-TISL but not TISL.

Proposition 16. LHOR *is not TISL.*

Proof. Fixing $k > 0$ and homomorphic tier τ, let us show that LHOR is not k-TISL on τ. Suppose $\mathrm{L} \notin \tau$. Then, observe that $\mathrm{suff}^{k-1}(\tau^{\leftarrow}(\lambda)) = \mathrm{suff}^{k-1}(\tau^{\leftarrow}(\mathrm{L})) = \rtimes$, but $\mathrm{LHOR}^{\rightarrow}_{\lambda}(\lambda) = \lambda$, while $\mathrm{LHOR}^{\rightarrow}_{\mathrm{L}}(\lambda) = \acute{\mathrm{L}}$. Thus, LHOR is not k-TISL on τ if $\mathrm{L} \notin \tau$. But if $\mathrm{L} \in \tau$, then we have $\mathrm{LHOR}^{\rightarrow}_{\mathrm{HL}^k}(\mathrm{H}) = \mathrm{H}$ and $\mathrm{LHOR}^{\rightarrow}_{\mathrm{L}^k}(\mathrm{H}) = \mathrm{L}\acute{\mathrm{H}}$ even though $\mathrm{suff}^{k-1}(\mathrm{HL}^k) = \mathrm{suff}^{k-1}(\mathrm{L}^k) = \mathrm{L}^{k-1}$. Therefore, LHOR is also not k-TISL on τ if $\mathrm{L} \in \tau$, so we conclude that it is not TISL. □

Proposition 17. LHOR *is* 2*-I-*2*-TISL.*

Proof. Consult Figure 1. □

Similarly, as Baek (2018) shows, the decision problem for LHOR is I-TSL but not TSL.

Proposition 18. C_{LHOR} *is not TSL.*

Proof. Fix $k > 0$, and suppose C_{LHOR} is k-TSL on homomorphic tier τ. It is clear that $\acute{\mathrm{H}}, \acute{\mathrm{L}} \in \tau$; otherwise, we would have $\acute{\mathrm{H}}\acute{\mathrm{H}}, \acute{\mathrm{L}}\acute{\mathrm{L}} \in C_{\mathrm{LHOR}}$. Furthermore, we must have $\mathrm{H} \in \Gamma$ and $\mathrm{L} \in \Gamma$, since otherwise we would have $\mathrm{H}\acute{\mathrm{H}} \in C_{\mathrm{LHOR}}$ and $\acute{\mathrm{L}}\mathrm{L} \in C_{\mathrm{LHOR}}$, respectively. Thus, every symbol of $\acute{\Sigma}$ is projected to tier τ, so C_{LHOR} is k-SL.

Now, let $S \subseteq (\acute{\Sigma}, \rtimes, \ltimes)^k$ be the set of substrings that are permitted to appear in strings of C_{LHOR}, and let $x := \mathrm{L}^k\acute{\mathrm{H}}\mathrm{L}^k$. Since $x \in C_{\mathrm{LHOR}}$, every length-k substring of $\rtimes^{k-1}x\ltimes^{k-1}$ is in S. However, observe that every length-k substring of $\rtimes^{k-1}xx\ltimes^{k-1} = \rtimes^{k-1}\mathrm{L}^k\acute{\mathrm{H}}\mathrm{L}^{2k}\acute{\mathrm{H}}\mathrm{L}^k\ltimes^{k-1}$ is also a substring of $\rtimes^{k-1}x\ltimes^{k-1}$, and is therefore also in S. Thus, we have deduced that $xx \in C_{\mathrm{LHOR}}$, contradicting the definition of C_{LHOR}. □

Proposition 19. C_{LHOR} *is* 2*-I-*2*-TSL.*

Proof. Let ψ be the homomorphic tier projection given by $\{\mathrm{H}, \acute{\mathrm{H}}, \acute{\mathrm{L}}\}$, and let τ be defined by

$$\tau(x) := \begin{cases} \psi(x)\mathrm{L}, & x \in \Sigma^*\mathrm{L} \\ \psi(x), & \text{otherwise.} \end{cases}$$

In other words, τ is the same as ψ, except the last symbol of the input is always projected. It is easy

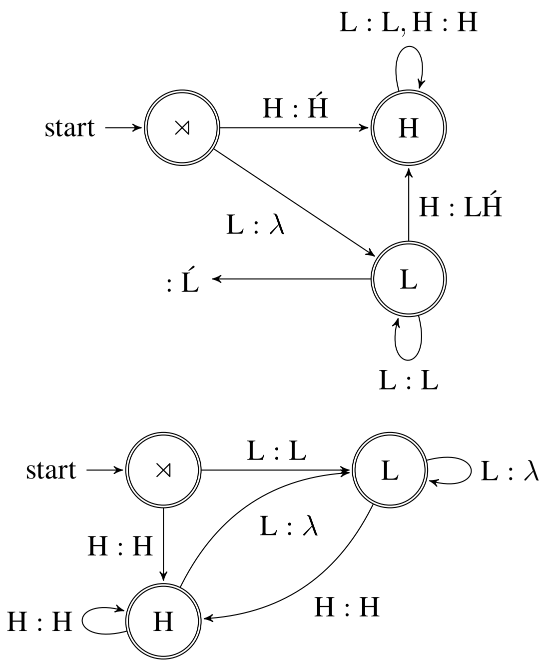


Figure 1: A 2-I-2-TISL SFST for LHOR (top) and a 2-ISL SFST for its tier projection (bottom).

to see that τ is 2-ISL. Now, observe that C_{LHOR} is 2-I-2-TSL on tier τ with permissible substrings ⋊H́, ⋊Ĺ, HH, HL, H⋉, H́H, H́L, H́⋉, Ĺ⋉, L⋉, and ⋊⋉. □

As these examples illustrate, the I-TISL functions and I-TSL languages form the current subregular complexity bounds for attested subsequential stress systems and their associated stress constraints (Baek, 2018; Hao and Andersson, 2019).[2] These results extend those of Heinz (2009), Rogers et al. (2013), and Heinz (2014), who observed that stress constraints belong to restrictive subclasses of the regular languages. Other ways of refining the subregular hierarchy for stress have been proposed; Rogers and Lambert (2019), for example, define the *strictly piecewise local* and the *piecewise locally testable* language classes. The remainder of this paper will seek to compare metrical grid theory against the benchmarks I-TISL and I-TSL benchmarks.

4 Metrical Grid Theory

Treatments of stress in phonological theory are typically based on the intuition that phonemes are organized into hierarchical structures, each level of which imposes prominence relations on its elements. While current approaches in Optimality Theory (OT) use constraints on the shapes of prosodic units and the prominence relations they impose (Prince and Smolensky, 1993, 2004; McCarthy and Prince, 1986, 1996, 1993), *metrical theory* has provided several frameworks for understanding stress outside of OT.[3] This section reviews and formalizes *metrical grid theory* (MGT, Prince, 1983; Halle and Vergnaud, 1987; Idsardi, 1992; Halle and Idsardi, 1995; Hayes, 1995), a classic example of such a framework.

According to MGT, prosodic relations are represented using diagrams like the ones that appear in Figure 2. Each syllable is associated with a continuous stack of asterisks. The height of each stack represents the prosodic prominence of its associated syllable, with the tallest stack marking primary stress and the second-tallest stack(s) marking secondary stress. Each layer of asterisks represents a level of the prosodic hierarchy: the bottom asterisks, the *syllable layer*, mark the location of each syllable; the middle asterisks, the *foot layer*, mark syllables that are prominent within their respective feet; and the top asterisk, the *word layer*, marks the syllable with the greatest prominence in the word.

		∗						Word
		∗	∗		∗		∗	Foot
∗	∗	∗∗	∗∗	∗	∗∗	∗	∗	Syllable
L	L	H́	H	L	H	L	L	

							∗	Word
							∗	Foot
∗	∗	∗	∗	∗	∗	∗	∗	Syllable
L	L	L	L	L	L	L	Ĺ	

Figure 2: Sample metrical grids for the LHOR system.

The placement of asterisks within the diagram is determined as follows. In the syllable layer, all light syllables receive an asterisk, while all heavy syllables receive two asterisks (∗∗). Thus, the syllable layer serves to record which syllables are heavy and which are light. In the foot layer, asterisks are placed by applying one or more of the following rules.

- **Quantity Sensitivity** (QS)**:** Place an asterisk directly above each ∗∗ in the syllable layer.

[2] Hao and Andersson (2019) and Koser and Jardine (To appear) show that some stress systems are not subsequential; I do not consider such systems in this paper.

[3] See Kager (1995) for a survey overview of various approaches in metrical theory.

- **Perfect Grid:** Place an asterisk in every second position, starting from the first (PG(odd)) or the second (PG(even)) position.
- **End Rule:** Place an asterisk in the first (ER(foot, $\lhd$)) or last (ER(foot, $\rhd$)) position.

In Figure 2, for example, foot-level asterisks are assigned according to QS and ER(foot, $\rhd$). Applying both rules means that an asterisk is added to a position if and only if *either* QS *or* ER(foot, $\rhd$) adds an asterisk to that position. Finally, the single word-layer asterisk is assigned according to the following End Rule.

- **End Rule:** Place an asterisk directly above the first (ER(word, $\lhd$)) or the last (ER(word, $\rhd$)) asterisk in the foot layer.

Using these rules, LHOR is implemented in MGT as follows: the foot-level asterisks are assigned using QS and ER(foot, $\rhd$), and the word-level asterisk is assigned using ER(word, $\lhd$). In words containing a heavy syllable, such as $\mathrm{L}^2\acute{\mathrm{H}}\mathrm{HLHL}^2$, the leftmost asterisk on the foot level occurs directly above the leftmost H in the word. Thus, ER(word, $\lhd$) assigns primary stress to the leftmost H. In words without a heavy syllable, such as L^8, QS does not place any asterisks on the foot layer, so the leftmost asterisk of the foot layer is the single asterisk placed by ER(foot, $\rhd$). This occurs at the right word boundary, so ER(word, $\rhd$) assigns primary stress to the rightmost syllable.

4.1 Formalizing MGT

Let us now give a precise definition of the system we have informally described. To represent stacks of asterisks, I annotate alphabet symbols with a subscript indicating the number of asterisks above that symbol. Since $**$ only occurs in the syllable layer directly above an H, I do not distinguish between a single position in the grid that contains $**$ and a single position that contains only one asterisk. For example, the upper grid in Figure 2 is represented by the string $\mathrm{L}_1^2\mathrm{H}_3\mathrm{H}_2\mathrm{L}_1\mathrm{H}_2\mathrm{L}_1\mathrm{L}_2$.

Definition 20. Let Σ be any alphabet, and for every $\sigma \in \Sigma$, let $\sigma_1, \sigma_2, \ldots$ be symbols not in $\acute{\Sigma}$. Let $\Sigma_0 := \Sigma$, and for $i \geq 0$, define the alphabet $\Sigma_i := \{\sigma_i | \sigma \in \Sigma\}$, with $\sigma_0 = \sigma$ for each $\sigma \in \Sigma$. Let $\Sigma_{\leq j} := \bigcup_{i=0}^{j} \Sigma_i$ and $\Sigma_* := \bigcup_{i=0}^{\infty} \Sigma_i$.

In this formalization, each rule must be associated with a particular level in the prosodic hierarchy. A rule associated with level i, where

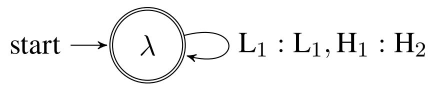


Figure 3: A homomorphic SFST implementing QS.

the syllable layer is level 1, takes as input a grid whose tallest stack of asterisks is at most i levels tall, and increments the height of stacks ending at the previous level by one.

Definition 21. For $i > 0$, a *level-i rule* is a same-length subsequential function $\rho : \Sigma_{\leq i}^* \to \Sigma_{\leq i}^*$ such that for each $x \in \Sigma_{\leq i}^*$ and for each position j,

- if $x[j] = \sigma_{i-1}$ for some $\sigma \in \Sigma$, then either $\rho(x)[j] = \sigma_{i-1}$ or $\rho(x)[j] = \sigma_i$;
- otherwise, $\rho(x)[j] = x[j]$.

Example 22. Figure 3 shows an SFST implementing QS as a level-2 rule. Since L_1 represents a column with a single asterisk and H_1 represents a column with $**$, this SFST simply changes all H_1s to H_2. ER($i, \lhd$) is represented by the following level-i rule:

$$\mathrm{ER}_i^{\lhd}(x) = \begin{cases} y\sigma_i z, & x = y\sigma_{i-1}z \text{ and } y \in \Sigma_{\leq i-2}^* \\ x, & \text{otherwise.} \end{cases}$$

ER($i, \lhd$) places an asterisk above the leftmost asterisk on level $i - 1$. Symbols in $\Sigma_i \cup \Sigma_{i-1}$ represent syllables with an asterisk on level $i - 1$; symbols in Σ_i represent syllables with an asterisk on both level $i - 1$ and level i. If the first symbol of x in $\Sigma_i \cup \Sigma_{i-1}$ is of the form $\sigma_{i-1} \in \Sigma_{i-1}$, then this symbol is incremented to σ_i. If the first symbol of x in $\Sigma_i \cup \Sigma_{i-1}$ is of the form $\sigma_i \in \Sigma_i$, then this symbol is left unchanged: ER($i, \lhd$) is still understood to add an asterisk on the ith level, but an asterisk has already been added there by another rule. If x does not contain any symbols of $\Sigma_i \cup \Sigma_{i-1}$, then ER($i, \lhd$) does not add any asterisks. Observe that $\mathrm{ER}_i^{\lhd}$ is 2-TISL on tier $\Sigma_i \cup \Sigma_{i-1}$.

The mapping of input words to their metrical-grid representations is simply the composition of a sequence of rules. Since rules can only place asterisks on top of existing asterisks from the previous layer, the rules in the sequence are required to be monotonically increasing in their associated level of the hierarchy.[4]

[4]This requirement is known in the phonological literature as the *continuous column constraint* (Hayes, 1995).

Definition 23. For $i > 0$, an *i-level metrical grid* is a function $\rho : \Sigma^* \to \Sigma^*_{\leq i}$ such that

$$\rho = \rho_n \circ \rho_{n-1} \circ \cdots \circ \rho_0$$

for some $n > 0$, where

- $\rho_0 : \Sigma^* \to \Sigma^*_1$ is the homomorphism given by $\rho(\sigma) = \sigma_1$ for all $\sigma \in \Sigma$;
- ρ_n is a level-i rule; and
- for all j, if ρ_j is a level-k rule, then ρ_{j+1} is either a level-k rule or a level-$(k+1)$ rule.

From an i-level metrical grid, we recover the stress system described by the grid by assuming that asterisks on level i represent primary stress.

Definition 24. Let ρ be an i-level metrical grid. The *stress system induced by* ρ is the stress system $s_\rho := s_i \circ \rho$, where $s_i : \Sigma^*_{\leq i} \to \acute{\Sigma}^*$ is the homomorphism given by

$$s_i(\sigma_j) := \begin{cases} \acute{\sigma}, & j = i \\ \sigma, & j < i. \end{cases}$$

5 Expressive Power of MGT

Trivially, the version of MGT formalized in Subsection 4.1 can express any subsequential stress system s: since level-i rules are allowed to be arbitrary subsequential functions, it suffices to construct a grid consisting of a level-2 rule that places an asterisk above the syllable assigned primary stress by s. In this section, I show that MGT is strictly more expressive than the I-TISL functions. The example that separates MGT from the I-TISL functions is motivated by Hao and Andersson's (2019) formalization of *Dybo's Rule* (Dybo, 1977), a description of unbounded stress in Abkhaz. I review Hao and Andersson's implementation of Dybo's Rule both as a stress system and as a 3-level grid in Subsection 5.1. In Subsection 5.2, I show that a slight modification of Hao and Andersson's stress system is in fact not I-TISL, even though the ability of MGT and I-TSL languages to describe the system is not affected by the change in representation, as will be shown in Subsection 5.3.

5.1 Dybo's Rule

In Abkhaz, syllables are lexically marked as being *dominant* or *recessive*. Dybo's Rule is an LHOR stress system in which dominant syllables not followed by other dominant syllables are considered to be heavy, and all other syllables are considered to be light. This is illustrated by the following examples, where dominant syllables are underlined.

(25) Dybo's Rule in Abkhaz (Spruit, 1986)

a. [dət͡ʃʰalaˈwama] 'Does (s)he usually go?'

b. [aˈʁʷakʼʲaməsa] '(the) poniard'

c. [apʰaˈra] 'to pleat'

d. [maaˈkʼə] 'one handle'

The dominant syllables [wa] in (25a), [ʁʷa] and [ma] in (25b), and [ra] in (25c) are heavy, since they are not followed by another dominant syllable. In (25a) and (25c), the sole heavy syllable receives primary stress. In (25b), the first of the two heavy syllables receives primary stress. (25d) does not have any heavy syllables, so the last syllable receives primary stress by default. Hao and Andersson (2019) represent Dybo's Rule using the following stress system.

Definition 26. Let $\Sigma := \{\mathrm{D}, \mathrm{R}\}$. The *two-letter Dybo's Rule* is the stress system $\alpha : \Sigma^* \to \acute{\Sigma}^*$ defined as follows. For $u \in \mathrm{R}^*\mathrm{D}^*$, $v \in \Sigma$, and $w \in \Sigma^*$,

$$\alpha(uvw) := \begin{cases} u\acute{v}w, & v = \mathrm{D} \text{ and } w \notin \mathrm{D}\Sigma^* \\ uw\acute{v}, & uvw \in \mathrm{R}^*. \end{cases}$$

In the two-letter Dybo's Rule, dominant syllables are represented by D and recessive syllables are represented by R. The first D not followed by an R receives primary stress. This stress system turns out to be an I-TISL function.

Proposition 27 (Hao and Andersson, 2019)**.** *The two-letter Dybo's Rule is* 2*-I-*3*-TISL.*

Hao and Andersson implement this system in MGT using a 3-level grid of the form $\mathrm{ER}^{\lhd}_3 \circ \rho_1 \circ \rho_0$, where ρ_0 is as defined in Definition 23 and ρ_1 is given by the 2-ISL SFST shown in the right panel of Figure 4. Following the MGT analysis of LHOR stress, ρ_1 serves to mark all heavy syllables, as well as the last syllable, with an asterisk on level 2. Thus, ρ_1 places an asterisk above all Ds followed by an R, along with the last syllable.

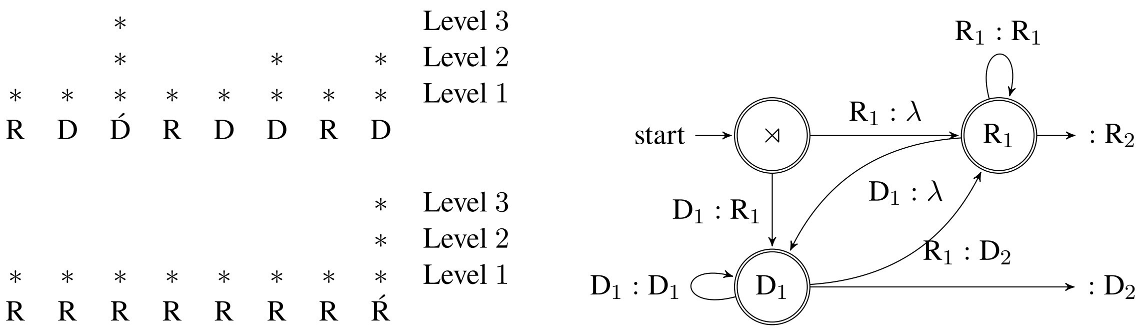


Figure 4: **Left:** Sample metrical grids for the two-letter Dybo's Rule. **Right:** 2-ISL SFST implementing the level-2 rule for the two-letter Dybo's Rule.

5.2 MGT vs. I-TISL Functions

Let us now introduce the following variant of the two-letter Dybo's Rule.

Definition 28. Let $\Sigma := \{\mathrm{D}, \mathrm{E}, \mathrm{R}\}$. The *three-letter Dybo's Rule* is the stress system $\delta : \Sigma^* \to \acute{\Sigma}^*$ defined as follows. For $u \in \mathrm{R}^*\{\mathrm{D}, \mathrm{E}\}^*$, $v \in \Sigma$, and $w \in \Sigma^*$,

$$\delta(uvw) := \begin{cases} u\acute{v}w, & v \neq \mathrm{R} \text{ and } w \notin \{\mathrm{D}, \mathrm{E}\}\Sigma^* \\ uw\acute{v}, & uvw \in \mathrm{R}^+. \end{cases}$$

The three-letter Dybo's Rule is exactly like the two-letter Dybo's Rule, except that there are two alphabet symbols representing dominant syllables: D and E. The MGT analysis of the two-letter Dybo's Rule can be easily adapted to the three-letter Dybo's Rule just by identifying E with D. The I-TISL implementation of the two-letter Dybo's Rule, however, cannot be applied to the three-letter Dybo's Rule.

Theorem 29. *The three-letter Dybo's Rule is not i-I-j-TISL on tier τ for any i, j, or τ.*

Proof. Suppose δ is i-I-j-TISL on tier τ. Observe that

$$\delta^{\rightarrow}_{\mathrm{E}^i\mathrm{D}}(\mathrm{R}) = \acute{\mathrm{D}}\mathrm{R} \qquad \delta^{\rightarrow}_{\mathrm{E}^i\mathrm{E}}(\mathrm{R}) = \acute{\mathrm{E}}\mathrm{R}$$
$$\delta^{\rightarrow}_{\mathrm{E}^i\mathrm{R}}(\mathrm{R}) = \mathrm{R} \qquad \delta^{\rightarrow}_{\mathrm{E}^i}(\mathrm{R}) = \acute{\mathrm{E}}\mathrm{R},$$

thus $\tau^{\leftarrow}(\mathrm{E}^i\mathrm{D})$, $\tau^{\leftarrow}(\mathrm{E}^i\mathrm{R})$, and $\tau^{\leftarrow}(\mathrm{E}^i)$ must all be distinct. Let $t := \tau^{\leftarrow}(\mathrm{E}^i)$, and for $\sigma \in \Sigma$, let t_σ be such that $\tau^{\leftarrow}(\mathrm{E}^i\sigma) = tt_\sigma$. Clearly, $t_\sigma \neq \lambda$ for every σ.

Let q_0 be the start state of the minimal SFST T for τ, let q be the state of T corresponding to $\tau^{\rightarrow}_{\mathrm{E}^i}$, let $\to$ be the transition function of T, and let r, y, and z be such that

$$q_0 \xrightarrow{\mathrm{E}^i:y} q \xrightarrow{\mathrm{DE}^i:z} r.$$

Since τ is i-ISL and

$$\mathrm{E}^i[: 1-i] = \mathrm{E}^i\mathrm{DE}^i[: 1-i] = \mathrm{E}^{i-1},$$

we must have $\tau^{\rightarrow}_{\mathrm{E}^i\mathrm{DE}^i} = \tau^{\rightarrow}_{\mathrm{E}^i}$, thus $r = q$. It follows that for every $k \geq 0$, $\tau^{\leftarrow}(\mathrm{E}^i(\mathrm{DE}^i)^k) = yz^k$. Since $t_\mathrm{E} \preceq z$, it must be the case that $|yz^k| \geq k$.

Now, observe that

$$\begin{aligned} &\tau^{\leftarrow}(\mathrm{DRE}^i(\mathrm{DE}^i)^j)[: 1-j] \\ &= \tau^{\leftarrow}(\mathrm{DRE}^i)z^j[: 1-j] \\ &= yz^j[: 1-j] \\ &= \tau^{\leftarrow}(\mathrm{E}^i(\mathrm{DE}^i)^j)[: 1-j]. \end{aligned}$$

Therefore, $\tau^{\rightarrow}_{\mathrm{DRE}^i(\mathrm{DE}^i)^j} = \tau^{\rightarrow}_{\mathrm{E}^i(\mathrm{DE}^i)^j}$. However, this contradicts the fact that

$$\tau^{\rightarrow}_{\mathrm{DRE}^i(\mathrm{DE}^i)^j}(\mathrm{R}) = \mathrm{R} \neq \acute{\mathrm{E}}\mathrm{R} = \tau^{\rightarrow}_{\mathrm{E}^i(\mathrm{DE}^i)^j}(\mathrm{R}),$$

so we conclude that δ is not i-I-j-TISL on tier τ for any i, j, or τ. □

The 2-I-3-TISL SFST given by Hao and Andersson (2019) for the two-letter Dybo's Rule projects DR sequences to the tier. When the SFST encounters a contiguous block of Ds, it must delay its output by one time step, as shown below, because it is unknown whether or not the current D should be assigned stress. Stress is not assigned until the SFST encounters an R or the end of the input string has been reached.

$$\rtimes\rtimes \xrightarrow{\mathrm{D}:\lambda} \rtimes\mathrm{D} \xrightarrow{\mathrm{D}:\mathrm{D}} \ldots \xrightarrow{\mathrm{D}:\mathrm{D}} \rtimes\mathrm{D} \xrightarrow{\mathrm{R}:\acute{\mathrm{D}}\mathrm{R}} \mathrm{DR}$$

Once the tier contains a full DR sequence, the SFST knows that stress has already been assigned, and therefore does not assign stress for the remainder of its computation.

With the three-letter Dybo's Rule, the state needs to record the identity of the most recent input symbol in order to delay the output by one time step. The only way to do this with an I-TISL SFST is to project the most recent input symbol to the tier.

$$\rtimes\rtimes \xrightarrow{\text{D}:\lambda} \rtimes\text{D} \xrightarrow{\text{E}:\text{D}} \text{DE} \xrightarrow{\text{D}:\text{E}} \text{ED} \to \ldots$$

Since an ISL tier projection cannot distinguish between the first block of dominant syllables in its input and subsequent blocks of dominant syllables, the schema shown above requires *every* block of dominant syllables to be projected to the tier. These syllable blocks overflow the memory provided by the tier, thus preventing it from recording whether or not stress has already been assigned.

5.3 MGT vs. I-TSL Languages

Despite the fact that the three-letter Dybo's Rule is not I-TISL, the stress constraint it induces is I-TSL.

Proposition 30. *C_δ is 2-I-3-TSL.*

Proof. Let τ be the 2-ISL tier projection that projects

- all instances of Ď, É, and Ŕ;
- all instances of DR, DŔ, ER, EŔ, ĎR, ĎŔ, ÉR, and ÉŔ; and
- the last symbol of the input.

Now, observe that C_δ is 2-I-3-TSL on tier τ, with the following permissible substrings: $\rtimes\rtimes\acute{\sigma}$, $\rtimes\rtimes\acute{\text{R}}$, $\rtimes\rtimes\ltimes$, $\rtimes\acute{\sigma}\text{R}$, $\rtimes\acute{\sigma}\ltimes$, $\rtimes\acute{\text{R}}\ltimes$, $\rtimes\ltimes\ltimes$, $\sigma\text{R}\gamma$, $\sigma\text{R}\ltimes$, $\text{R}\sigma\text{R}$, $\text{R}\ltimes\ltimes$, $\acute{\sigma}\text{R}\gamma$, $\acute{\sigma}\text{R}\ltimes$, $\acute{\sigma}\ltimes\ltimes$, and $\acute{\text{R}}\ltimes\ltimes$, where $\sigma, \gamma \in \{\text{D}, \text{E}\}$. □

The tier projection described here is similar to the tier projection used for Hao and Andersson's (2019) 2-I-3-TISL implementation of α. Like the 2-I-2-TSL grammar for C_{LHOR}, the 2-I-3-TSL grammar for C_δ projects all heavy syllables and stressed syllables to the tier, along with the last syllable of the input. Unlike the grammar for C_{LHOR}, the grammar for C_δ also projects recessive syllables following dominant syllables. This allows the grammar to ensure that all stressed dominant syllables are dominant: they must be immediately followewd by either R or $\ltimes$.

Because neither an ITSL grammar nor a metrical grid needs to produce the surface form as output, the problem of using the tier to delay computation does not arise for the ITSL implementation of C_δ or for the MGT analysis of δ. While there is still a discrepancy between layer 2 of the MGT analysis and the tier projection used for C_δ, I conjecture based on this observation that MGT describes ITSL decision problems.

Conjecture 31. *Let $\rho = \text{ER}_3^{\lhd} \circ \rho_1 \circ \rho_0$ be a 3-level metrical grid. If ρ_1 is ISL, then C_{s_ρ} is I-TSL.*

6 Conclusion

In comparing the I-TISL implementation of α with the MGT analysis, Hao and Andersson (2019) express the intuition that generalized tier projections and MGT are similar in that both systems use intermediate representations in order to compute stress. The analysis of Section 5 has revealed that this similarity is superficial because the computations carried out by I-TISL functions, I-TSL languages, and metrical grids are fundamentally different from one another. The most prominent of the differences discussed here is that systems implementing the transduction problem need to transfer a substantial amount of information about the underlying form to the surface form, while systems implementing the decision problem only need to retain enough information to distinguish a grammatical string from an ungrammatical one. Thus, the transduction problem may be viewed as conceptually more difficult than the decision problem.[5] While metrical grids compute transductions, their memory capabilities are enhanced by the fact that rule composition allows state information to be encoded in intermediate layers. Conjecture 31 suggests that this enhanced memory may be sufficient for MGT to bridge the gap between the transduction problem and the decision problem.

In conclusion, the comparison of generalized tier projection with metrical grids provides an instructive example of an analytical tool—intermediate representations of prominence relations—that behaves differently depending on the formalism in which it is instantiated. This approach could potentially offer a way to compare different theoretical frameworks in terms of how they accommodate superfically similar proposals. I leave the exploration of such ideas to future work.

[5] This asymmetry mirrors the relationship between *search problems* and *decision problems* in computational complexity theory (see Arora and Barak, 2009, pp. 54–55 for an overview). The conjecture that NP $\supsetneq$ P captures the intuition that the search problem is the more difficult one.

References

Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge, United Kingdom.

Emmon Bach. 1975. Long Vowels and Stress in Kwakiutl. *Texas Linguistic Forum*, 2:9–19.

Hyunah Baek. 2018. Computational representation of unbounded stress: Tiers with structural features. In *Proceedings of CLS 53 (2017)*, volume 53, pages 13–24, Chicago, IL. Chicago Linguistic Society.

Franz Boas, Helene Boas Yampolsky, and Zellig S. Harris. 1947. Kwakiutl Grammar with a Glossary of the Suffixes. *Transactions of the American Philosophical Society*, 37(3):203–377.

Phillip Burness and Kevin McMullin. 2019. Efficient learning of Output Tier-based Strictly 2-Local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. PhD Dissertation, University of Delaware, Newark, DE.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output Strictly Local Functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*, pages 112–125, Chicago, IL. Association for Computational Linguistics.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*, 1 edition. Harper & Row, New York, NY.

Vladimir Antonovič Dybo. 1977. Западнокавказская акцентная система и проблема ее происхождения. In *Конференция «Ностратические языки и ностратическое языкознание»: Тезисы докладов*, pages 41–45, Moscow, Soviet Union. USSR Academy of Sciences.

John Anton Goldsmith. 1976. *Autosegmental Phonology*. PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Morris Halle and William J. Idsardi. 1995. General Properties of Stress and Metrical Structure. In John A. Goldsmith, editor, *The Handbook of Phonological Theory*, 1 edition, Blackwell Handbooks in Linguistics, pages 403–443. Wiley-Blackwell, Cambridge, MA.

Morris Halle and Jean-Roger Vergnaud. 1987. *An Essay On Stress*. Number 15 in Current Studies in Linguistics. MIT Press, Cambridge, MA.

Yiding Hao and Samuel Andersson. 2019. Unbounded Stress in Subregular Phonology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 135–143, Florence, Italy. Association for Computational Linguistics.

Bruce Hayes. 1995. *Metrical Stress Theory: Principles and Case Studies*. University of Chicago Press, Chicago, IL.

Jeffrey Heinz. 2009. On the role of locality in learning stress patterns. *Phonology*, 26(2):303–351.

Jeffrey Heinz. 2014. Culminativity times harmony equals unbounded stress. In Harry van der Hulst, editor, *Word Stress: Theoretical and Typological Issues*, pages 255–275. Cambridge University Press, Cambridge, United Kingdom.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, number 23 in Phonology and Phonetics, pages 126–195. De Gruyter Mouton, Berlin, Germany.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based Strictly Local Constraints for Phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 58–64, Portland, OR. Association for Computational Linguistics.

William James Idsardi. 1992. *The Computation of Prosody*. PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

René Kager. 1995. The Metrical Theory of Word Stress. In John A. Goldsmith, editor, *The Handbook of Phonological Theory*, 1 edition, Blackwell Handbooks in Linguistics, pages 367–402. Wiley-Blackwell, Cambridge, MA.

Nate Koser and Adam Jardine. To appear. Stress assignment and subsequentiality. In *Proceedings of the 2019 Annual Meeting on Phonology*, Proceedings of the Annual Meetings on Phonology, Stony Brook, NY. Linguistic Society of America.

Connor Mayer and Travis Major. 2018. A Challenge for Tier-Based Strict Locality from Uyghur Backness Harmony. In *Formal Grammar 2018, 23rd International Conference, FG 2018, Sofia, Bulgaria, August 11-12, 2018, Proceedings*, volume 10950 of *Lecture Notes in Computer Science*, pages 62–83, Berlin, Germany. Springer Berlin Heidelberg.

John J. McCarthy and Alan Prince. 1986. Prosodic Morphology. Technical Report 32, Rutgers Center for Cognitive Science, New Brunswick, NJ.

John J. McCarthy and Alan Prince. 1993. Generalized alignment. In Geert Booij and Jaap Van Marle, editors, *Yearbook of Morphology 1993*, Yearbook of Morphology, pages 79–153. Springer Netherlands, Dordrecht, Netherlands.

John J. McCarthy and Alan Prince. 1996. Prosodic Morphology 1986. Submission 13, University of Massachusetts Amherst Department of Linguistics, Amherst, MA.

Adam McCollum, Eric Baković, Anna Mai, and Eric Meinhardt. 2018. The expressivity of segmental phonology and the definition of weak determinism. *LingBuzz*, lingbuzz/004197.

Alan Prince and Paul Smolensky. 1993. Optimality Theory: Constraint Interaction in Generative Grammar. Technical Report 2, Rutgers University, New Brunswick, NJ.

Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing, Malden, MA.

Alan S. Prince. 1983. Relating to the Grid. *Linguistic Inquiry*, 14(1):19–100.

George N. Raney. 1958. Sequential Functions. *Journal of the Association for Computing Machinery*, 5(2):177–180.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and Sub-regular Complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108, Berlin, Germany. Springer Berlin Heidelberg.

James Rogers and Dakotah Lambert. 2019. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

Jacques Sakarovitch. 2009. *Elements of Automata Theory*. Cambridge University Press, Cambridge, United Kingdom.

Arie Spruit. 1986. *Abkhaz Studies*. PhD Dissertation, Leiden University, Leiden, Netherlands.

Odile Vaysse. 1986. Addition molle et fonctions p-locales. *Semigroup Forum*, 34(1):157–175.

The Subregular Complexity of Syntactic Islands

Nazila Shafiei and **Thomas Graf**
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
nazila.shafiei@stonybrook.edu

Abstract

We provide a formal framework for analyzing syntactic island effects from a subregular perspective. Key aspects of the syntactic representation are encoded as strings where precedence represents containment. Island effects then are expressed as constraints on the shape of these strings. The constraints fit in the class IBSP (Interval-Based Strictly Piecewise), which has been previously explored in subregular phonology. Consequently, the characterization of islands in terms of IBSP string constraints not only provides a computational upper bound on the inventory of feasible island effects, but also establishes a surprising link between syntax on the one hand and phonology on the other.

1 Introduction

The subregular program is concerned with analyzing the complexity of linguistic dependencies that are at most regular. The program has found great success in computational phonology (see Heinz 2018 and references therein), where it has resulted in a computational typology of phonological patterns and corresponding learning algorithms. Syntax, by virtue of being mildly context-sensitive, may seem far beyond the purview of the subregular program. But syntax is also subregular once one considers more suitable representations. Two routes have been explored: lifting subregular classes from strings to trees (Graf, 2018b; Vu et al., 2019), and putting string constraints on particular path languages of syntactic trees (Graf and Shafiei, 2019). Whereas the former has been mostly used in the analysis of structure building operations, the latter has been applied to syntactic constraints such as NPI-licensing.

This paper focuses on an area where these two aspects of syntax meet: island constraints. Island constraints impose additional restrictions on displacement, which in the tradition of Transformational grammar is equated with the operation *Move*. The shape of islands is narrowly circumscribed, indicating that they are very limited from a computational perspective. In this paper, we confirm this intuition. Island constraints are expressed as constraints over a path language where linear precedence in the string encodes (a specific notion of) containment. Given such a string representation, island constraints fall into the subregular class Interval-Based Strictly Piecewise (IBSP), which has been argued to play a central role in phonology (Graf, 2017, 2018a). At the same time, IBSP is sufficiently weak to rule out many unattested island constraints. Our paper thus makes several contributions: it deepens our understanding of subregular syntax, establishes parallels to phonology, and provides linguists with a computational theory of islands.

Due to space constraints, we focus largely on strong islands, and only on the canonical cases for most of them. We also investigate the *that*-trace constraint and the coordinate structure constraint, and we show that they cannot be handled in the system proposed here. This paper thus marks but the first step towards a fully articulated, empirically grounded theory of islands.

The discussion proceeds as follows: the preliminaries section (§2) discusses Minimalist grammars (§2.1), our string representation format (§2.2), and the subregular class IBSP (§2.3). Section 3 presents the central result that a number of (strong) island constraints follow a uniform IBSP pattern of very low complexity. We start with the adjunct island constraint (§3.1) and then generalize the analysis to wh-islands, the complex np constraints, the subject condition, and freezing effects (§3.2). Section 4 then explores the limits of IBSP over a-strings. On the one hand this allows us to correctly rule out many unattested island constraints, but it also means that the approach cannot handle all aspects of the *that*-trace constraint and the coordinate structure constraint. In addition, our approach currently lacks any notion of linguistic naturalness, which allows for some very odd (albeit computationally simple) island constraints (§5).

2 Preliminaries

The paper rests on several research traditions, which are briefly sketched in this section: Minimalist grammars as a formal model of syntax (§2.1), string representations for syntax (§2.2), and the subregular class of IBSP string languages (§2.3).

2.1 Minimalist Grammars

Since island constraints have mostly been studied in the generative tradition, we adopt Minimalist grammars (MGs; Stabler, 1997) as a formal model of syntax. MGs are a derivational grammar formalism for building tree structures by combining feature-annotated lexical items via the operations Merge and Move. Figure 1 gives a concrete example of this process. Only a few key aspects of MGs matter for this paper, in particular their feature system (see Stabler 2011 for a full discussion).

Each lexical item consists of a phonetic exponent and a string of features. There are four distinct types of features. *Category features* (X^-) and *selector features* (X^+) establish head argument relations via *Merge*. The other two feature types drive the operation *Move*. A *licensee feature* f^- indicate that the phrase headed by the lexical item undergoes f-movement, and the matching *licensor feature* f^+ indicates the landing site of f-movement. As in Minimalist syntax, movement is a mechanism for displacing subtrees of an already assembled tree, and movement always targets the closest available landing site (encoded in MGs via licensor features).

Given the special role of adjuncts in island constraints, we also adopt the adjunction mechanism of (Frey and Gärtner, 2002). Instead of a category feature, a lexical item l may carry an *adjunction feature* $\mathrm{X}^\sim$ which allows it to adjoin to an XP.

An MG's structure building process is usually represented as a derivation tree like the one in Fig. 1. But we will frequently represent derivation trees with the more compact format of dependency trees. The rightmost tree in Fig. 1 presents a concrete example.

2.2 String Representations for Syntax

Our investigation of island constraints will not operate directly over trees, but rather over strings that represent specific aspects of the tree structure. This follows recent work by Graf and Shafiei (2019), who analyze syntactic constraints such as NPI-licensing and Principle A as operating over strings that encode asymmetric c-command relations. A tree is well-formed iff it holds for every node n in the tree that the relevant string representation for n is well-formed with respect to the syntactic string constraints.

Graf and Shafiei (2019) choose a string representation that encodes both containment and a limited form of c-command (cf. Frank and Vijay-Shanker, 2001). These *augmented command strings* (or simply *c-strings*) can be defined in various ways, but the easiest option uses MG dependency trees. We adopt this definition but simplify it so that the resulting string representation only keeps track of containment. For this reason, we call these strings *ancestor strings* (or simple *a-strings*).

Definition 1 (A-strings). Let t be an MG dependency tree. If n is the root of t, then $\mathrm{as}(n) := n$. If n has mother m, then $\mathrm{as}(n) := n\ \mathrm{as}(m)$. ⌟

Example. In Fig. 1, $\mathrm{as}(\text{Mary} :: \mathrm{D}^-\mathrm{nom}^-) = \text{Mary} :: \mathrm{D}^-\mathrm{nom}^-\ \text{buy} :: \mathrm{D}^+\mathrm{D}^+\mathrm{V}^- \uparrow \varepsilon :: \mathrm{V}^+\mathrm{nom}^+\mathrm{T}^- \uparrow \text{did} :: \mathrm{T}^+\mathrm{wh}^+\mathrm{C}^-$. For increased readability, we may omit features and replace empty heads by their category. Then $\mathrm{as}(\textit{Mary}) =$ *Mary buy T did.* ⌟

The use of strings is a matter of mathematical convenience. The results obtained this way can be backported to subregular machinery that operates directly on dependency trees or derivation trees (Graf and De Santo, 2019). This will be discussed further in §5.

2.3 Subregular Complexity

Formal language theory has a rich tradition of studying proper subclasses of the regular string languages (McNaughton and Papert, 1971; Pin, 1997; Yli-Jyrä, 2005, a.o.). More recently, this line of work has been picked up and extended by computational phonologists (see Heinz 2018 and references therein). The class Interval-Based Strictly Piecewise (IBSP) was proposed as a linguistically natural unification of previously pro-

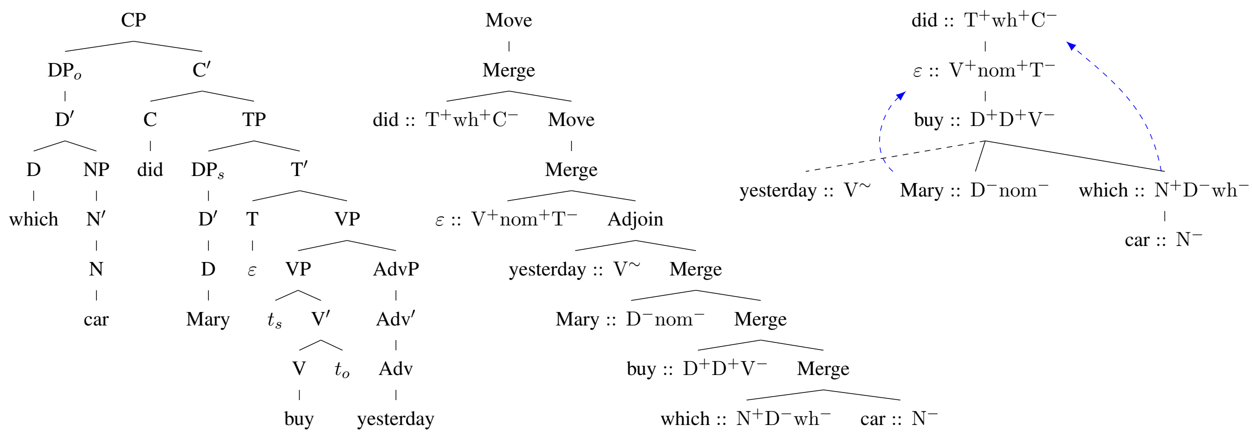


Figure 1: X′-tree, MG derivation tree, and equivalent dependency tree for *Which car did Mary buy yesterday*

posed classes for subregular phonology (Graf, 2017, 2018a). IBSP constitutes an approximate upper bound on string dependencies in phonology.

Definition 2 (k-val). A *segmented k-interval* ($k \geq 0$) over alphabet Σ, or simply *segmented k-val*, is a tuple $\langle L, R, F_i \rangle_{0 \leq i \leq k}$ such that

- $L, R \subseteq \Sigma \cup \{\varepsilon\}$ specify the *left edge* and *right edge*, respectively, and
- $F_i \subseteq \Sigma$ specifies the i-th *filler* slot. ⌟

Definition 3 (IBSP-k). Let Σ be some fixed alphabet and $\rtimes, \ltimes \notin \Sigma$ two distinguished symbols. An IBSP-k grammar over Σ ($k \geq 0$) is a pair $G := \langle i, S \rangle$, where i is a segmented k-val over $\Sigma \cup \{\rtimes, \ltimes\}$ and $S \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a set of forbidden k-grams. A string $s \in \Sigma^*$ is generated by G iff there is no k-gram $u_1 \cdots u_k \in S$ such that $\rtimes^k s \ltimes^k$ is a member of the language

$$(\Sigma \cup \{\rtimes, \ltimes\})^* \cdot L \cdot F_0^* \cdot \{u_1\} \cdot F_1^* \cdot \{u_2\} \cdot \ldots \cdot F_{k-1}^* \cdot \{u_k\} \cdot F_k^* \cdot R \cdot (\Sigma \cup \{\rtimes, \ltimes\})^*$$

The language $L(G)$ is the set of all $s \in \Sigma^*$ that are generated by G. A stringset L is IBSP-k iff $L = L(G)$ for some IBSP-k grammar G. ⌟

In the definition above, * represents the usual Kleene closure. The symbol $\cdot$ denotes string concatenation, lifted to sets: $A \cdot B := \{ab \mid a \in A, b \in B\}$.

Following Graf and Shafiei (2019), we can use IBSP grammars over strings to regulate the shape of trees.

Definition 4 (IBSP over trees). Let G be an IBSP grammar, t an MG dependency tree, and f_t a total function from nodes of t to strings. Then t is well-formed with respect to G iff it holds for all lexical items l in t that $f_t(l)$ is generated by G. ⌟

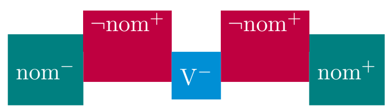
Example. Suppose that nom-movement is forbidden out of VPs. Over a-strings, this corresponds to the requirement that no V-head may occur between a node with nom$^-$ and the closest node with nom$^+$ (because movement in MGs always targets the closest head with a matching feature). This can be expressed as the following IBSP-1 grammar:

$$\begin{aligned}
L &:= \{l \mid l \text{ carries nom}^-\} \\
F_0 &:= \{l \mid l \text{ does not carry nom}^+\} \\
F_1 &:= \{l \mid l \text{ does not carry nom}^+\} \\
R &:= \{l \mid l \text{ carries nom}^+\} \\
S &:= \{l \mid l \text{ carries V}^-\}
\end{aligned}$$

Then as(*Mary*) $=$ *Mary buy T did* will be deemed illicit because it matches a forbidden pattern with L := *Mary*, $F_0^* := \varepsilon$, $F_1^* := \varepsilon$, R := *T*, and *buy* $\in S$. Consequently, the dependency tree is not well-formed, either. ⌟

It is often convenient to represent IBSP grammars in a more visual format. The example grammar above corresponds to the diagram below.

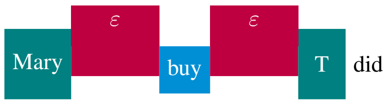
(1) *Graphical representation of an IBSP grammar*



The outermost vertical boxes represent the left and right edge, respectively. The square in the middle represents a position of the forbidden k-grams

— since the example grammar uses forbidden unigrams, there is only one such square. The vertically offset boxes represent the fillers, in this case F_0 and F_1. We use features to as a shorthand for the set of lexical items that carry this feature. For instance, nom^- denotes the set of all lexical items carrying nom^-. The expression $\neg\mathrm{nom}^+$ denotes the of all lexical items that do not carry the relevant feature, in this case nom^+. This visual format can also be used to show that a string is ill-formed.

Example. Recall that $\mathrm{as}(\mathit{Mary}) = \mathit{Mary\ buy\ T\ did}$ is illicit. We can show this by giving a specific instantiation of the interval and the k-gram in the string.



This disagram conveys the same information as the formal description in the previous example. ⌟

In the next section, we use this machinery to analyze syntactic island effects from a subregular perspective. We show that strong islands follow a fixed IBSP pattern over a-strings that is exceedingly simple.

3 Strong Islands over A-Strings

The notion of syntactic islands originates from Transformational Grammar (Ross, 1967). From the perspective of MGs, a constituent C is an island iff no phrase contained by C may have a licensee feature checked by a matching licensor feature outside C. A distinction is commonly made between *strong islands* and *weak islands*. Strong islands limit movement irrespective of whether the mover is an argument or an adjunct. Weak islands, on the other hand, limit adjunct movement but not argument movement. We will focus mostly on strong islands in this paper. We first analyze the adjunct island constraint (§3.1) as an IBSP perspective over a-strings, and then show how the same template can be used for several other strong island effects (§3.2).

3.1 Adjunct Island Constraint

The adjunct island constraint is arguably the most robust case of a strong island. It is illustrated by the contrast in (2).

(2) *Adjunct island constraint*

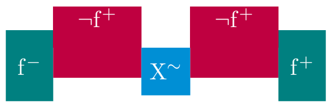
a. Which car did John complain $[_{\mathrm{CP}}$ that he can't fix _]?

b. * Which car did John complain $[_{\mathrm{CP}}$ because he can't fix _]?

In both (2a) and (2b) the wh-phrase *which car* moves out of an embedded clause. But in (2a) the embedded clause is an argument of the verb *complain*, whereas it is an adjunct in (2b). That movement is allowed out of the argument clause but not the adjunct clause is referred to as the adjunct island constraint.

The adjunct island constraint can be easily expressed as an IBSP constraint on a-strings. In fact, it uses the template we already encountered in §2.3, except that the set of forbidden unigrams consists of all adjuncts rather than all verbs. As explained in §2.1, we adopt the proposal of Frey and Gärtner (2002) that every adjunct carries an adjunct feature $\mathrm{X}^{\sim}$ that allows it to adjoin to XPs. The IBSP grammar for the adjunct island constraint thus corresponds to the following template.

(3) *IBSP-1 grammar for adjunct islands*



Note that this template actually represents multiple IBSP grammars as f must be correctly instantiated for each movement feature: nom for subject movement, wh for wh-movement, top for topicalization, and so on. If all of those were put inside a single IBSP grammar, then one lose the fact that the left edge and the right edge must be opposite polarities of the same feature — a nom^- for the left edge could be paired up with an wh^+ as the right edge. Since IBSP lacks a direct means of coordinating left and right edges like this, we instead have to posit a separate grammar for each movement feature in order to correctly enforce the adjunct island constraint for that specific movement type.

Example. Figure 2 shows the MG dependency tree for *which topic did you leave because Mary talked about* _. This sentence contains illicit wh-movement out of an adjunct. Now consider the a-string for *which*, with the relevant features indicated in square brackets: $\mathrm{as}(\mathit{which}) = \mathit{which}[\mathrm{wh}^-]$ *talked about T because*$[\mathrm{V}^{\sim}]$ *leave T did*$[\mathrm{wh}^+]$. As shown by the diagram below, this a-string is ill-formed with respect to the IBSP grammar in (3) (assuming $\mathrm{f} := \mathrm{wh}$).

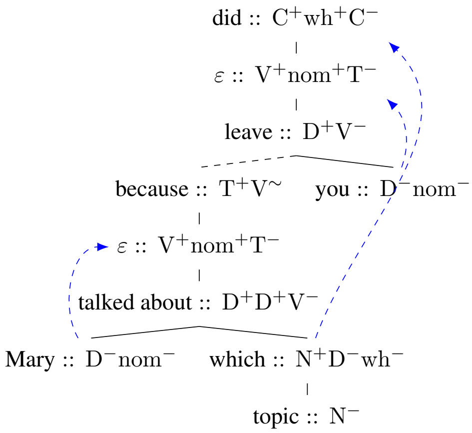
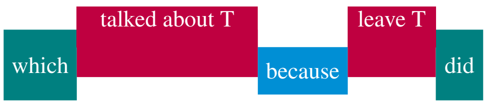


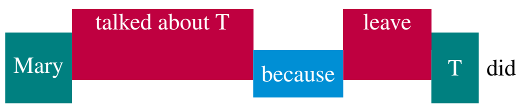
Figure 2: Adjunct island violation



As the a-string is illicit, the whole sentence is ill-formed. ⌟

The reader may wonder why the template explicitly forbids f^{+} as fillers. This ensures that the right edge is always the closest f^{+}, which is the one targeted for movement by the lexical item with f^{-} in the left edge. Without this restriction, the IBSP grammar would incorrectly rule out well-formed movement patterns.

Example. Consider once more the example sentence *which topic did you leave because Mary talked about _* as depicted in Fig. 2. This sentence contains two instances of nom-movement, both of which are well-formed. But now consider the IBSP grammar regulating nom-movement. Suppose that this grammar allows for lexical items with nom^{+} to appear in the filler slots. Then this grammar would incorrectly rule out $\mathrm{as}(\textit{Mary}) = \textit{Mary}[\mathrm{nom}^{-}]$ *talked about* $T[\mathrm{nom}^{+}]$ *because*$[\mathrm{V}^{\sim}]$ *leave* $T[\mathrm{nom}^{+}]$ *did*.



⌟

The reader should also keep in mind that the use of $\mathrm{X}^{\sim}$ is just a notational shorthand for specifying a list of lexical items. One can remove some items from this set to allow for exceptions to the adjunct island constraint, such as the ones noted by Truswell (2007).

(4) a. * Which car did John drive Mary crazy [while he tried to fix _]?

b. Which car did John drive Mary crazy [while trying to fix _]?

Assuming a distinction between finite T-heads (T^{-}) and other T-heads (T^{-}_{inf}), we can account for this by excluding while :: $\mathrm{T}^{+}_{inf}\mathrm{V}^{\sim}$ from the list of forbidden lexical items.

In sum, the adjunct island constraint can be handled by a very simple and intuitive IBSP grammar (or rather, a collection of such grammars, one for each movement type). From a formal perspective, this IBSP grammar looks very similar to the IBSP treatment of blocking effects in phonology. In phonology, an intervening consonant cluster may block long-distance harmony. In syntax, an intervening head with an adjunction feature interrupts the dependency between an f^{-} and an f^{+}. The existence of the adjunct island constraint thus becomes a bit less mysterious: it is very simple from a computational perspective, and it employs a general blocking mechanism that also seems to be active in other parts of language.

3.2 Other Strong Islands

Besides the adjunct island constraint, the class of strong islands also includes wh-islands, complex NPs, and subjects. The corresponding constraints are illustrated below.

(5) *Wh-island constraint*

a. Which movie did John say that Mary liked _?

b. * Which movie did John wonder whether Mary liked _?

(6) *Complex NP constraint*

a. What did you say [that John bought _]?

b. * What did you hear rumors [that John bought _]?

(7) *Subject condition*

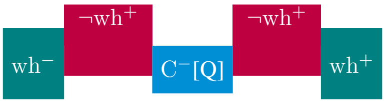
a. Who did John write [a story about _]?

b. * Who was [a story about _] written by John?

These all use minor variations of the template for the adjunct island constraint.

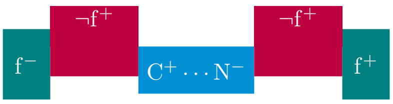
Let us start with the wh-island constraint. Here it suffices to make two changes. Since most types of movements, e.g. topicalization, are not affected by this constraint, we limit the possible instantiations for f^- and f^+ to just wh^- and wh^+, respectively. Then the list of blockers is changed from adjuncts to all elements that induce wh-islands. These are commonly taken to be all C-heads that have some kind of question semantics, including *whether*, *how*, and *if*. We denote this set $\mathrm{C}^-[\mathrm{Q}]$.

(8) *IBSP-1 grammar for wh-islands*



Next we turn to the complex NP constraint. This one, too, uses the basic template of the adjunct island constraint, but we once again have to change the list of blockers. In the complex NP constraint, the blocking is not done by an adjunct, but by a more complex structural configuration: movement out of a CP is illicit if the CP is the argument of a noun. Thanks to the MG feature calculus, we can rephrase this as a ban against moving out of an NP that selects a CP,[1] which means that the set of blockers contains all lexical items, and only those, that contain a selector feature C^+ and a category feature N^-. We denote this set of lexical items by $\mathrm{C}^+ \cdots \mathrm{N}^-$.

(9) *IBSP-1 grammar for complex NP constraint*



The reader is invited to verify that this grammar correctly rules out the sentence *what did you hear rumors that John bought*, which is depicted in Fig. 3.

This leaves us with the subject condition, which can actually be regarded as an instance of what is known as *freezing effects*. This describes the phenomenon that once a phase XP has undergone movement, it becomes opaque to extraction. Any mover inside XP has to move out of the phrase before it starts moving. From the perspective of MGs, this can be rephrased as a constraint on the distribution of movement features. Let f_1^- and g_1^-

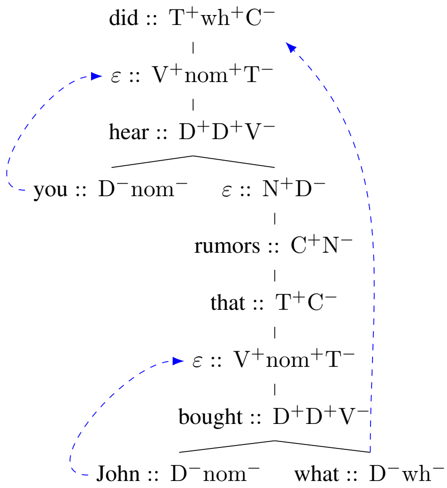
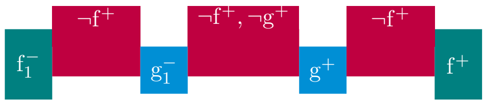


Figure 3: Violation of the complex NP constraint

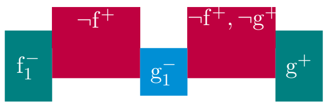
denote lexical items whose first movement feature is f^- and g^-, respectively. If the phrase headed by g_1^- contains f_1^-, then the target of f_1^- must be contained by the target of g_1^-. We can capture this generalization by moving from an IBSP-1 grammar to an IBSP-2 grammar (or rather, a collection of such grammar for every possible choice of f^- and g^-).

(10) *IBSP-2 grammar for freezing effects*

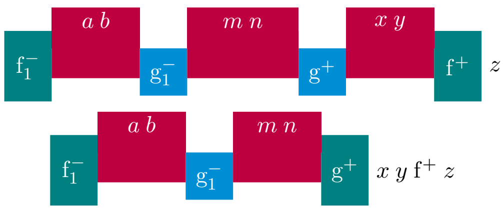


The step up from IBSP-1 to IBSP-2 makes freezing effects appear more complex. But it is actually possible to get the same effect just with an IBSP-1 grammar. The trick is to make g^+ the right edge of the k-val rather than f^+.

(11) *IBSP-1 grammar for freezing effects*



Example. Consider the abstract a-string $\mathrm{f}_1^-\ a\ b\ \mathrm{g}_1^-\ m\ n\ \mathrm{g}^+ x\ y\ \mathrm{f}^+\ z$. Both grammars correctly rule it out as illicit.



Similarly, both grammars agree that the minimally different $\mathrm{f}_1^-\ a\ b\ \mathrm{g}_1^-\ m\ n\ \mathrm{f}^+ x\ y\ \mathrm{g}^+\ z$ is well-formed. ⌟

[1]Our feature-based interpretation of the complex NP constraint is actually stronger than the original version. Suppose that the NP selects a CP as its complement and some XP as its specifier. The complex NP constraint as originally stated would allow the XP to be extracted, whereas our version does not. As far as we have been able to determine, there are no nouns that take two arguments in this configuration, let alone one where the XP then is allowed to undergo movement.

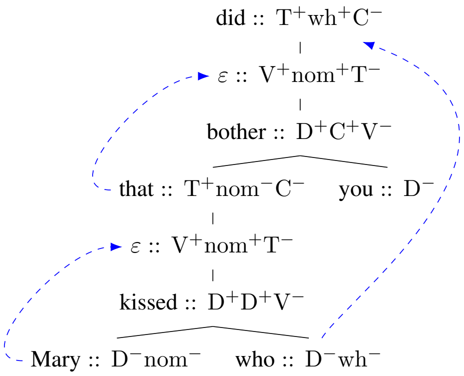
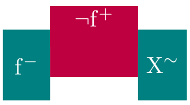


Figure 4: Violation of the subject condition

Both grammars also agree that the tree in Fig. 4 is illicit because of the ill-formed a-string of *who*.

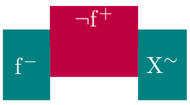
Note that we can apply the same kind of truncation strategy to the IBSP grammars for the other island constraints. This effectively reduces their complexity of IBSP-1 to IBSP-0. As laid out in Def. 3, an IBSP-0 grammar consists only of the left edge L, the right edge R, and a single filler F_0 inbetween. The set of forbidden k-grams is immaterial as every string is ruled out that matches $(\Sigma \cup \{\rtimes, \ltimes\})^* \cdot L \cdot F^* \cdot R \cdot (\Sigma \cup \{\rtimes, \ltimes\})^*$.

(12) *IBSP-0 grammar for adjunct islands*



(13) *IBSP-0 grammar for wh-islands*

(14) *IBSP-0 grammar for complex NP constraint*



In sum, all four island constraints can be captured with very simple IBSP-1 grammars (or even IBSP-0 grammars) over a-strings. Adjunct islands, wh-islands, and the complex NP constraint all follow the very same pattern. Subject islands, as a specific subcase of freezing effects, have a slightly higher complexity in that they are either IBSP-2 or IBSP-1. This depends on whether one requires the left and right edge of the k-val to be tied to the same feature f. Since freezing effects are widely considered to be more complex than standard island constraints and depend on the interaction of multiple movements, it is unsurprising that their IBSP complexity should be slightly higher. Nonetheless the IBSP approach with a-string provides a unified perspective of several movement restrictions that highlights their computational simplicity and treats them as a natural syntactic counterpart of blocking effects in phonology.

4 The Limits of A-Strings

The previous section has argued that IBSP grammars over a-strings provide an insightful perspective on movement constraints that highlights their simplicity and their formal parallels to blocking effects in phonology.

It is also noteworthy just how limited the machinery is. For instance, it is now unsurprising that no language has island constraints such as "you may move out of as many adjuncts as you have movement features". This simply cannot be expressed with IBSP-1 or IBSP-0. Similarly, we correctly predict that no language has complex structural conditions like "an adjunct is an island iff it is c-commanded by another adjunct". Not only would this require a larger k-val than IBSP-1 and IBSP-0 provide, the use of a-strings makes it completely impossible to refer to c-commanders. By adopting a string representation that only keeps track of containment, c-command conditions become inexpressible. While every Y in $\mathrm{as}(X)$ c-commands X, not every c-commander of X appears in $\mathrm{as}(X)$ — only those that are heads of phrases containing X do so. The absence of some c-commanders in a-strings thus makes them unsuitable to express c-command conditions.

The limits of a-strings with respect to c-command is both a curse and a blessing. As just discussed, it has the advantage of greatly limiting the predicted typology of island constraints. At the same time, it also means that the current approach is entirely incapable of handling some well-known restrictions on movement: the *that*-trace effect, and the coordinate structure constraint.

Let us first consider the *that*-trace effect, the core cases of which are illustrated below:

(15) a. Who do you think [Mary will leave _]?

b. Who do you think [_ will leave Mary]?

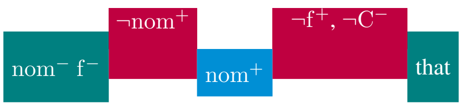
c. Who do you think [that Mary will leave _]?

d. * Who do you think [that _ will leave Mary]?

The *that*-trace filter forbids a subject to move across the head of the smallest containing sentential CP if that head is empty. This adds several new complications, but these can all be handled with IBSP.

The restriction to subjects amounts to the requirement that the left edge of the k-val must be a mover whose first movement feature is nom^-, followed by some f^-. Similarly, the limitation to sentential CPs can be expressed in terms of the MG feature calculus. The complementizer in the examples above has the feature make-up $\mathrm{T}^+\mathrm{C}^-$, whereas the complementizer of a relative clause, for instance, would have $\mathrm{T}^+\mathrm{N}^\sim$ (under an analysis of relative clauses as NP-adjuncts; other analyses require different features, but it will never be $\mathrm{T}^+\mathrm{C}^-$). So this aspect of the *that*-trace effect does not challenge the IBSP perspective either. Finally, the requirement that the constraint only applies to the closest such complementizer can be captured by restricting the appropriate filler. Overall, the typical instances of the *that*-trace constraint can be handled by an IBSP-1 grammar that uses the same truncation trick as our IBSP-1 treatment of freezing effects in (11).

(16) *IBSP-1 grammar for the* that*-trace effect*



For the core cases, then, the *that*-trace effect exceeds the strong island constraints in complexity, but is comparable to freezing effects.

However, there are cases where *that*-trace violations are repaired, and these cannot be handled in our approach. For instance, the *that*-trace effect does not apply when the gap is c-commanded by additional material.

(17) Who do you think [that [under no circumstances] _ will leave Mary]?

Here *under no circumstances* is an adjunct that attaches to TP or some other position below the complementizer and above the subject gap. This adjunct does not contain the gap, it only c-commands it. As a result, it is not present in the relevant a-strings, which makes it impossible for us to suspend the *that*-trace constraint. In order to handle this case, one needs a representation that encodes both c-command and containment, e.g. the c-strings of Graf and Shafiei (2019).

But the addition of c-command actually undermines the whole approach because it becomes impossible to determine a mover's landing site. Recall that our grammars block f^+ from occurring in the fillers so that we can correctly pick out the landing site for f-movement, i.e. the closest containing head with f^+. Crucially, heads that c-command the mover but do not contain it are not viable landing sites. For instance, if we are looking at the c-string of some f-mover that is the complement of some head H, the specifier of H may be some lexical item carrying f^+. This specifier should be allowed to go into a filler slot. At the same time, a c-commander that both carries f^+ and contains the f-mover should not be allowed to go into a filler slot. Since fillers are specified as lists of lexical items, there is no way to distinguish in their specification between containing c-commanders and all other c-commanders. Either we allow both in the filler or neither, and in each case we end up with an unsuitable grammar. IBSP is too weak to make the relevant distinctions with a representation format that encodes both c-command and containment.

While *that*-trace repair points out a limitation of IBSP, the coordinate structure constraint challenges the very notion of string-based representations for movement. This constraint forbids extraction from a conjunct, except if movement takes place across-the-board from all conjuncts.

(18) a. * Which wine did [Ed brew beer and Greg drink _]?

b. Which wine did [Ed brew _ and Greg drink _]?

Since there is no c-command or containment relation between the gaps in (18b), neither one appears in the other's c-string or a-string. Consequently, the c-strings for the object of *drink* do not differ at all between the two sentences, which makes it impossible to give a c-string account of this island constraint irrespective of how powerful one's computational apparatus is.

These two constraints show that IBSP over a-strings does not provide a fully exhaustive theory of islands or movement constraints. But the IBSP approach does highlight the structural uniformity of many islands, their computational simplicity, and their parallels to blocking effects in phonology. While our findings are still preliminary and need to be vetted by detailed analysis of a much wider range of constraints across many languages, it is encouraging that they closely mirror previous findings in phonology and yield rigorous claims about the possible shapes of islands.

5 Linguistic Naturalness

The previous section focused on some shortcomings of our approach with respect to expressivity, but there is also the issue of linguistic naturalness. First, the choice of string representations is unusual. Second, the reliance on lists of lexical items for specifying the components of an IBSP grammar means that there is no notion of naturalness. We acknowledge both issues, but we think that they can be insightfully addressed in future work.

As was briefly mentioned in §2.2, a-strings are just a convenient abstraction and the findings of this paper can be restated in terms of formal machinery that operates over trees instead of strings. This includes the tree tiers of Graf (2018b) and the sensing tree automata of Graf and De Santo (2019). But in both cases the necessary math is more likely to obfuscate the simplicity of the underlying principles, and the use of tree structures hides that the simple notion of containment is already enough to state many conditions on movement. We thus maintain that a-strings are methodologically useful even if they may not be cognitively real.

This leaves the lack of natural classes. It is true that our current approach is still too lenient a characterization of the class of possible island constraints. For instance, one can easily write an IBSP grammar over a-strings that does not allow topicalization across a ditransitive verb. Similarly, the ability to account for some exceptions such as (4) also allows us to specify ludicrous exceptions, for instance that the head of an adjunct induces an island unless it is a palindrome. These are clearly undesirable options, but they are typical of computational work. Our primary goal was to analyze island constraints from a subregular perspective to more accurately pinpoint their overall complexity. This allows us to put an upper bound on what island constraint may look like, but this is still a very generous bound. The formal restrictions must be paired with a theory of linguistic substance to accurately circumscribe the class of possible island constraints (see e.g. Graf 2013 for one such account for the adjunct island constraint).

6 Conclusion

We have argued that the most common cases of strong islands can be expressed as IBSP-1 (or IBSP-0) constraints on string representations that encode only containment. This formal characterization establishes new parallels to phonology and tightens the linguistic typology by excluding logically conceivable yet unattested island constraints. While a lot of empirical modeling work remains to be done, we are confident that this novel perspective on islands will prove very fertile.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. We would like to thank the three anonymous reviewers for their detailed comments, which prompted a major rewrite of the paper.

References

Robert Frank and K Vijay-Shanker. 2001. Primitive c-command. *Syntax*, 4(3):164–204.

Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in Minimalist grammars. In *Proceedings of the Conference on Formal Grammar*, pages 41–52.

Thomas Graf. 2013. The syntactic algebra of adjuncts. In *Proceedings of CLS 49*. To appear.

Thomas Graf. 2017. The power of locality domains in phonology. *Phonology*, 34:385–405.

Thomas Graf. 2018a. Locality domains and phonological c-command over strings. In *NELS 48: Proceedings of the Forty-Eighth Annual Meeting of the North East Linguistic Society*, volume 1, pages 257–270, Amherst, MA. GLSA.

Thomas Graf. 2018b. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, pages 117–136.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press, Cambridge, MA.

Jean-Eric Pin. 1997. Syntactic semigroups. In *Handbook of Language Theory*, pages 679–764. Springer, Berlin.

John R. Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Robert Truswell. 2007. Extraction from adjuncts and the structure of events. *Lingua*, 117:1355–1377.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.

Anssi Yli-Jyrä. 2005. *Contributions to the Theory of Finite-State Based Grammars*. Ph.D. thesis, University of Helsinki.

Multi-Input Strictly Local Functions for Templatic Morphology

Hossep Dolatian and Jonathan Rawski
Dept. of Linguistics
Institute for Advanced Computational Science
Stony Brook University
{hossep.dolatian,jonathan.rawski}@stonybrook.edu

Abstract

This paper presents an automata-theoretic characterization of templatic morphology. We generalize the Input Strictly Local class of functions, which characterize a majority of concatenative morphology, to consider multiple lexical inputs. We show that strictly local asynchronous multi-tape transducers successfully capture this typology of nonconcatenative template filling. This characterization and restriction uniquely opens up representational issues in morphological computation.

1 Introduction

Recent work in mathematical phonology connects phonological mappings to subclasses of the regular functions (McNaughton and Papert, 1971; Rogers and Pullum, 2011; Rogers et al., 2013; Heinz and Lai, 2013; Chandlee, 2014). One of the simplest subclasses is the Input Strictly Local (ISL) functions, which take as input a single string and generate an output based on local information. Despite their reduced expressivity, ISL functions capture an overwhelming majority of phonological and morphological maps (Chandlee, 2017; Chandlee and Heinz, 2018). In addition, ISL functions are provably easier and faster to learn than full regular functions (Chandlee et al., 2015a).

In this paper, we generalize this notion of locality from the above single-input functions to functions which take *multiple* strings as input. Such functions are called *Multi-Input Strictly Local* (MISL). MISL functions are computed by deterministic asynchronous Multi-tape Finite State Transducers (MT-FSTs). Natural language has processes which are understood in terms of enriched multi-string input structures, i.e. autosegmental structure. We focus on root-and-pattern (RPM) morphology or template-filling in Semitic. This paper shows that when formalized as a multi-input function, most RPM patterns are MISL.

Semitic RPM has has often been computed using different types of of MT-FSTs. By showing that that the bulk of Semitic RPM can be computed with *only* MISL MT-FSTs, this can act as a stepping stone to determining the learnability of RPM. It likewise acts as a benchmark to examine the typology of attested and unattested RPM processes. Furthermore, by using multi-input functions with MT-FSTs instead single-input functions with FSTs, we can more iconically compute the fact that 1) RPM consists of separate tiers for roots, inflection, and templates, and that 2) this separation makes certain RPM processes be local.

Single-input functions are a special case of multi-input functions. With finite-state calculus, single-input functions correspond to rational functions when they are modeled with 1-way single-tape FSTs, and to regular functions when modeled by 2-way single-tape FSTs (Filiot and Reynier, 2016).[1] Multi-input functions correspond to the class of functions modeled by 1-way or 2-way MT-FSTs. Although there is work on the expressivity of MT-FSTs (Furia, 2012), little is known on multi-input *functions* and their algebra, expressivity, and hierarchy (Frougny and Sakarovitch, 1993). We show that a locally defined subclass, MISL, carves a substantial chunk of Semitic RPM.

2 Preliminaries

2.1 Preliminaries for single-input functions

Let $\rtimes$,$\ltimes$be the start and end boundaries respectively. Let Σ be a finite alphabet of symbols (excluding $\rtimes$,$\ltimes$). Let $\Sigma_{\rtimes} = \Sigma \cup \{\rtimes, \ltimes\}$. Let Σ^* the set of all strings over Σ. Let $|w|$ indicate the length of $w \in \Sigma^*$. For two strings w and v let wv be their

[1] By single-tape FST, we mean a two-tape FST with one input tape and one output tape. Note that the functions computed by 1-way FSTs are called ‘regular functions’ in American computer science. In this paper, we follow French conventions which call this class the ‘rational functions’ (Filiot and Reynier, 2016).

concatenation, and for a set $L \subset \Sigma^*$ of strings and a string w, by wL we denote $\{wv | v \in L\}$. Let λ denote the empty string.

Given some string u and a natural number k, the *k-suffix* of u is the last k symbols of u: $\texttt{suff}(u, k) = v$ s.t. $|v| = k$ and $xv = u$ for some $x \in \Sigma^*$. For an alphabet Σ, the *k-factors* of Σ are the set of strings $w \in \Sigma*$ such that $|w| \leq k$.

Informally, a single-input function f is k-ISL if for all $u_1, u_2 \in \Sigma^*$, if $\texttt{suff}(u_1, k-1) = \texttt{suff}(u_2, k-1)$ then the two strings have the output extensions w.r.t f (Chandlee, 2014; Chandlee et al., 2015b). For any k-ISL function f over domain Σ^*, there exists a *canonical* deterministic single-tape finite-state transducer (1T-FST) M such that $|M| = f$ (meaning M computes f), and every state $q \in Q$ in M is labelled with one of the $k-1$ suffixes of Σ^*. Transitions are function tuples $\Delta : Q \times \Sigma \rightarrow Q \times \Gamma^*$. For a state $q \in Q$ and input symbol $a \in \Sigma$, $\delta(q, a) = (p, B)$ such that $B \in \Gamma^*$ and $p = \text{suff}(qa, k-1)$.

2.2 Preliminaries for multi-input functions

We introduce notation for functions which take multiple strings as input. To do so, we use tuples demarcated by brackets. In the formalization here, we only consider functions which produce one output string, not a tuple of output strings. But extending the formalization is trivial; such a function is illustrated in another paper of ours in the same volume.

A function f is an n-input function if it takes as input a tuple of n strings: $[w_1, \ldots, w_n]$, which we represent as $\vec{w}$, where each word w_i is made up of symbols from some alphabet Σ_i such that $w_i \in \Sigma_i^*$. Each alphabet Σ_i may be disjoint or intersecting, so two input strings w_i, w_j may be part of the same language Σ_i^*. These n alphabets form a tuple $\vec{\Sigma}$. Tuples can be concatenated: if $\vec{w} = [ab, c], \vec{x} = [d, ef]$, then $\vec{w}\vec{x} = [abd, cef]$.

To generalize the notion of suffixes into multiple strings, we define a tuple of n natural numbers as $\vec{k} = [k_1, \ldots, k_n]$. Given some tuple of n strings $\vec{w}$ and tuple of n numbers $\vec{k}$, $\vec{k}$*-suffix* of $\vec{w}$ is a tuple $\vec{v}$ of n strings v_i, made up of the last k_i symbols on w_i: $\texttt{suff}(\vec{w}, \vec{k}) = V$ s.t. $\vec{v} = [v_1, \ldots, v_n]$ and $|v_i| = k_i$ and $x_i v_i = w_i$ for $x_i \in \Sigma_i^*$. E.g. for $\vec{w}$=[abc,def] and $\vec{k} = [2, 1]$, $\texttt{suff}(\vec{w}, \vec{k}) = [bc, f]$. Given a tuple $\vec{k}$, the operation $\vec{k} - x$ subtracts x from each of k_i. E.g., for $\vec{k} = [2, 3, 6]$, $\vec{k} - 1 = [1, 2, 5]$. For a tuple of alphabets $\vec{\Sigma}$, the $\vec{k} - factors$ of $\vec{\Sigma}$ is the set of tuples $\vec{w} \in \vec{\Sigma}$ such that $|w_i| \leq k_i$. For example with

Let f be an n-input function defined over an $n-$tuple $\vec{w}$ of input strings $\vec{w} = [w_1, \ldots, w_n]$ taken from the tuple of n alphabets $\vec{\Sigma}$. As an *informal* and intuitive abstraction from ISL functions, f is Multi-Input Strictly Local (MISL) for $k = [k_1, \ldots, k_n]$ if the function operates over a bounded window of size k_i for w_i. Formally,

Definition 1: A function f is $\vec{k}$-MISL iff there exists a deterministic asynchronous Multi-tape FST such that i) $|M| = f$, and ii) the MT-FST is canonically $\vec{k}$-MISL

We explain $\vec{k}$-MISL Multi-tape FSTs in the next section.

Definition 1 is a automata-based definition of an MT-FST. We are currently working on finding a language-theoretic-based definition of an MISL function. Possible definitions for ISL functions, such as the use of tails or output extensions, cannot be easily extended to MISL functions. This is because are functions which have an MISL MT-FST, *but* the function has an infinite set of tails. We are currently investigating whether a monoidal definition of MISL functions is useful.

For an ISL function, it does not matter if the input string is read left-to-right or right-to-left. But for an MISL function, it does. A function may be left-to-right MISL but not right-to-left MISL. We leave out a proof but an illustration is given in another paper of ours in the same volume.

2.3 Multi-tape finite-state transducers

Multi-input functions can be modeled by multi-tape FSTs (MT-FST). An MT-FST is conceptually the same as single-tape FSTs, but over *multiple* input tapes (Rabin and Scott, 1959; Elgot and Mezei, 1965; Fischer, 1965; Fischer and Rosenberg, 1968; Furia, 2012). MT-FSAs and MT-FSTs are equivalent, and single-tape FSTs correspond to an MT-FSA with two tapes.

Informally, a MT-FST reads n multiple input strings as n input tapes, and it writes on a single output tape. Each of the n input strings is drawn from its own alphabet Σ_i. The output string is taken from the output alphabet Γ. For an input tuple of n strings $\vec{w} = [w_1, \ldots, w_n] = [\sigma_{1,1} \ldots \sigma_{1,|w_1|}, \ldots, \sigma_{n,1} \ldots \sigma_{n,|w_n|}]$, the initial configuration is that the MT-FST is in the initial state q_0, the read head. The FST begins at the first position of each of the n input tapes $\sigma_{i,1}$, and the writing head of the FST is positioned at the beginning of an empty output tape. After the FST reads the symbol under the read head, three things occur: 1) the state changes; 2) the FST writes some string; 3) the read head may advance to the right (+1) or stay put (0) on different tapes: either move on all tapes, no tapes, or some subset of the tapes.

This process repeats until the read head "falls off" the end of each input tape. If for some input $\vec{w}$, the MT-FST falls off the right edge of the n input tapes when the FST is in an accepting state after writing u on the output tape, we say the MT-FST transduces, transforms, or maps, $\vec{w}$ to u or $f_T\vec{w} = u$.[2] Otherwise, the MT-FST is undefined at $\vec{w}$. We illustrate MT-FSTs in §4.

Formally, a $n-$MT-FST for some natural number n is a 6-tuple $(Q, \vec{\Sigma_{\rtimes}}, \Gamma, q_0, F, \Delta)$ where:

- n is the number of input tapes
- Q is the set of states
- $\vec{\Sigma_{\rtimes}} = [\Sigma_{1\rtimes}, \ldots, \Sigma_{n\rtimes}]$ is a tuple of n input alphabets Σ_i which include the end boundaries $\Sigma_{i\rtimes}$
- Γ is the output alphabet
- $q_0 \in Q$ is the initial state
- $F \subset Q$ is the set of final states
- $\delta : Q \times \vec{\Sigma_{\rtimes}} \to Q \times \vec{D} \times \Gamma^*$ is the transition function where
  - $D = \{0, +1\}$ is the set of possible directions,[3]
  - $\vec{D} = [D^n]$ is an n-tuple of possible directions to take on each tape

The above definition can be generalized for MT-FSTs which use multiple output tapes. As parameters, an MT-FST can be deterministic or non-deterministic, synchronous or asynchronous. We only use *deterministic* MT-FSTs which are weaker than non-deterministic MT-FSTs. An MT-FST is synchronous if all the input tapes are advanced at the same time, otherwise it is asynchronous. We use asynchronous MT-FSTs which are more powerful than synchronous MT-FSTs. Synchronous MT-FSTs are equivalent to multi-track FSAs which are equivalent to single-tape FSAs, making them no more expressive than regular languages. For a survey of the properties of MT-FSAs and MT-FSTs, see Furia (2012).

[2]If the MT-FST generates tuples instead of single strings, then the MT-ST maps $\vec{w}$ to $\vec{u}$.

[3]If the MT-FST reads from right to left, then it uses the -1 direction parameter

A configuration c of a $n-$MT-FST M is an element of $(\vec{\Sigma_{\rtimes}}^* Q \vec{\Sigma_{\rtimes}}^* \times \Gamma^*)$, short for $([\Sigma^*_{1\rtimes} q \Sigma^*_{1\rtimes}, \ldots, \Sigma^*_{n\rtimes} q \Sigma^*_{n\rtimes}] \times \Gamma^*)$. The meaning of the configuration $c = ([w_1 q x_1, \ldots, w_n q x_n], u)$ is the following. The input to M is the tuple $\vec{w}\vec{x} = [w_1 x_1, \ldots, w_n x_n]$. The machine is currently in state q. The read head is on each of the n-input tapes on the first symbol of x_i (or has fallen off the right edge of the input tape if $x_i = \lambda$). u is currently written on the output tape.

Let the current configuration be $([w_1 q a_1 x_1, \ldots, w_n q a_n x_n], u)$ and let the current transition arc be $\delta(q, [a_1, \ldots, a_n]) = (r, \vec{D}, v)$. If $\vec{D} = [0^n]$, then the next configuration is $([w_1 r a_1 x_1, \ldots, w_n r a_n x_n], uv)$ in which case we write $([w_1 q a_1 x_1, \ldots, w_n q a_n x_n], u) \to ([w_1 r a_1 x_1, \ldots, w_n r a_n x_n], uv)$ (= none of the tapes are advanced) . If $\vec{D} = [+1^n]$, then the next configuration is $([w_1 a_1 r x_1, \ldots, w_n a_n r x_n], uv)$ in which case we write $([w_1 q a_1 x_1, \ldots, w_n q a_n x_n], u) \to ([w_1 a_1 r x_1, \ldots, w_n a_n r x_n], uv)$ (= all the tapes are advanced). Otherwise, the next configuration is $([w_i C_1 x_1 \ldots, w_n C_n x_n, \ldots], uv)$ where $C_i = r a_i$ if $D_i = 0$ and $C_i = a_i r$ if $D_i = +1$ in which case we write $([w_1 q a_1 x_1, \ldots, w_n q a_n x_n], u) \to ([w_i C_1 x_1 \ldots, w_n C x_n, \ldots], uv)$ (= a subset of the tapes are advanced).[4]

The transitive closure of $\to$ is denoted with $\to^+$. Thus, if $c \to^+ c'$ then there exists a finite sequence of configurations $c_1, c_2 \ldots, c_n$ with $n > 1$ such that $c = c_1 \to c_2 \to \ldots \to c_n = c'$.

As for the function that a MT-FST M computes, for each $n-$tuple $\vec{w} \in \vec{\Sigma}^*$ where $\vec{w} = [w_1, \ldots, w_n]$, $f_M(\vec{w}) = u \in \Gamma^*$ (where $f_M = |M|$) provided there exists $q_f \in F$ such that $([q_0 \rtimes w_1 \ltimes, \ldots, q_0 \rtimes w_n \ltimes], \lambda) \to^+ ([\rtimes w_1 \ltimes q_f, \ldots, \rtimes w_n \ltimes q_f], u)$. Otherwise, if the configuration is $([\rtimes w_1 \ltimes q, \ldots, \rtimes w_n \ltimes q], u)$ and $q \notin F$ then the transducer crashes and the transduction f_T is undefined on input $\vec{w}$. Note that if a MT-FST is deterministic, it follows that if $f_T(\vec{w})$ is defined then u is unique.

As explained in §2.2, we define a function as $\vec{k}$-MISL iff there exists a corresponding deterministic asynchronous $\vec{k}$-MISL Multi-tape FST.

Definition 2: A deterministic asynchronous MT-FST M with alphabet $\vec{\Sigma}$ is a canonical MT-

[4]Note that the interpretation of the third type of configuration subsumes the first two. We explicitly show the first two for illustrative reasons.

FST for an $\vec{k}$-MISL function f if the states of M are labelled with the $\vec{k}-1$ suffixes of $\vec{\Sigma}$.

In Definition 2, the restriction on state labels does not apply to the unique initial state and unique final state. In other words, except for the initial and final states q_0 and q_f, every state corresponds to a possible $\vec{k}-1$ *factor* of f

.

3 Root-and-pattern morphology in template filling

Semitic root-and-pattern morphology (RPM) involves segmenting a word into multiple discontinuous morphemes or morphs: a consonantal root **C**, inflectional vocalism **V**, and prosodic template **T**.[5] A partial paradigm of Standard Arabic verbs is in Table 1, amassed from McCarthy (1981). To illustrate, the verb *kutib* (Table 1a) is morphologically composed of a root **C**=*ktb*, vocalism **V**=*ui*, and template **T**=*CVCVC* which marks locations for consonants and vowels. Its autosegmental structure is provided in Table 1a.[6]

The bulk of theoretical and psycholinguistic results show that Semitic RPM *does* involve template-filling (Prunet, 2006; Aronoff, 2013; Kastner, 2016), but the formulation of templates is controversial (Ussishkin, 2011; Bat-El, 2011). One hypothesis is that the template is composed of CV slots (McCarthy, 1981). Alternatives are that the template is made of prosodic units like moras, syllables, and feet (McCarthy and Prince, 1990a,b), is derived from other templates via affixation (McCarthy, 1993), or is a set of optimized prosodic constraints (Tucker, 2010; Kastner, 2016; Zukoff, 2017). Alternatively, the job of the template is done by deriving words from other words via *overwriting* or changing the vowels and consonants (Ussishkin, 2005), e.g. *katab+ui→kutib*.

We take a theory-neutral position and focus on the *mathematical* function behind RPM. Mathematically, RPM is a 3-input function that takes as input a 3-tuple $\vec{w} = [w_1, w_2, w_3]$ where w_1 is the root **C**, w_2 is the vocalism **V**, w_3 is the template **T**. The input alphabets are $\Sigma_1 = \Sigma_C$ of consonants, $\Sigma_2 = \Sigma_V$ of vowels, and $\Sigma_3 = \Sigma_T$ of prosodic slots {C,V} and other elements (moras, affixes). Each alphabet includes the start and end boundaries $\rtimes, \ltimes$: $\Sigma_{i\rtimes} = \Sigma_i \cup \{\rtimes, \ltimes\}$. The output alphabet is the output segments.

Thus mathematically, many of the formalizaitons of templates are equivalent. Whether the template or **T**-string is made from CV units or moras is a *notational* difference (Kiraz, 2001) and does not affect locality. The use of derivational affixation is analogous to function composition; it does not affect locality and is discussed in §4.1.3,§4.2. For prosodic optimization, the function still needs to be well-defined over multiple inputs and this makes a template be implicitly present in the function. This is discussed in (Dolatian and Rawski, 2019). As for an overwriting approach, it still requires a mechanism for placing the new segments that references discontinuity. That is, the function *katab+ui→kutib* implicitly assumes that the vowels can be separated: *kVtVb+ui→kutib*. The fact that one of the inputs is a template with filled consonants *kVtVb* can be equally well broken down to a root and template *ktb+CVCVC*.

Computationally, different models have been used to compute the above *mathematical* function *behind* Semitic RPM: single-tape FSTs (Bird and Ellison, 1994; Beesley and Karttunen, 2000, 2003; Cohen-Sygal and Wintner, 2006; Roark and Sproat, 2007), synchronous MT-FSAs (Kiraz, 2000, 2001; Hulden, 2009), and non-deterministic asynchronous MT-FSTs (Kay, 1987; Wiebe, 1992). For a review, see Kiraz (2000, 92), Kiraz (2001, Ch4),and Wintner (2014, 47). We model RPM with asynchronous deterministic MT-FSTs in order to capture its locality properties, which we explain next.

4 Multi-Input Locality in Semitic

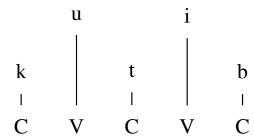
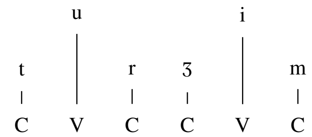
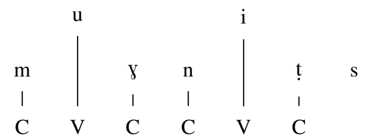
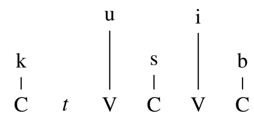
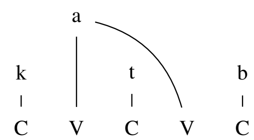
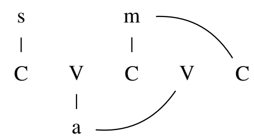
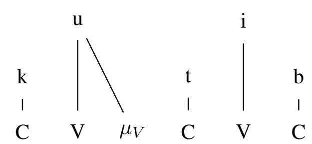
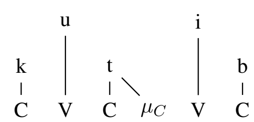
Mathematically, there is little discussion on the locality or non-locality of RPM. Chandlee (2017) shows that template-filling cannot be easily modeled with single-tape FSTs without sacrificing locality. Although not ISL, we show that the majority of RPM processes in Table 1 are MISL.

Arabic roots are generally at most 5 segments, vocalisms at most 2 segments, and the template is at most 12 slots (McCarthy, 1981). With this

[5] In Hebrew, some roots consists of consonants *and* vowels (Kastner, 2016). This difference is computationally trivial as long the template still treats Cs and Vs differently.

[6] We do not formalize RPM functions in broken plurals (Hammond, 1988; McCarthy and Prince, 1990b). Kiraz (2001, 106) formalizes it as a MT-FSA which use two inputs tapes: the singular and the vocalism. The singular tape can be annotated with prosodic information. We conjecture that broken plural formation is also MISL because there are no long-distance dependencies. We leave out a full formalization for space.

Table 1: Partial paradigm of Arabic root-and-pattern morphology with stable $\vec{k}$-values.

	Slot-filling pattern	Binyan	Gloss	Output	Root	Vowels	Template	k-value
a	1-to-1	Measure I Passive	*kutib*	‘was written’	*ktb*	*ui*	*CVCVC* 	[1,1,1]
b	... four consonants	Measure QI Passive	*turʒim*	‘was translated’	*trʒm*	*ui*	*CVCCVC* 	[1,1,1]
c	... with final deletion	Borrowed verb	*maɣnaṭ*	‘be magnetized’	*mɣnṭs*	*ui*	*CVCCVC* 	[1,1,1]
d	... with pre-association	Measure VIII Passive	*k<t>usib*	‘was gained’	*ksb*	*ui*	*CtVCVC* 	[1,1,1]
e	1-to-many... ... final spread of... ... vowels	Measure I Active	*katab*	‘it wrote’	*ktb*	*a*	*CVCVC* 	[1,2,1]
f	... consonants	Measure I Active	*samam*	‘he poisoned’	*sm*	*a*	*CVCVC* 	[2,1,1]
g	... medial spread of... ... (long) vowels	Measure III Passive	*kuutib*	‘be corresponded’	*ktb*	*ui*	$CV\mu_V CVC$ 	[1,2,1]
h	... (geminate) consonants	Measure II Passive	*kuttib*	‘be caused to write’	*ktb*	*ui*	$CVC\mu_C VC$ 	[2,1,1]

bound, RPM is reducible to modeling a function over a finite domain and range, i.e., a *finite* list of input-output pairs. Throughout this section, we abstract away from this. Our functions assume that there is no bound on the size of the root **C**, vocalism **V**, or template **T**. This allows us to treat RPM as a function over an infinitely sized domain. Doing so allows us to better capture the underlying function's generative capacity (Savitch, 1993). See (Dolatian and Rawski, 2019) for details on the role of infinity in computing Semitic RPM.

4.1 1-to-1 slot-filling

4.1.1 Simple 1-to-1 slot-filling

For *kutib* (Table 1a), RPM shows 1-to-1 slot-filling, meaning the e association of segments on any two strings is 1-to-1. The number of vowels in the vocalism **V** match the number of *V* slots in the template **T**. The same applies for the number of consonants in the root **C** and the *C* slots in **T**.

1-to-1 slot-filling is [1,1,1]-MISL or MISL for $\vec{k} = [1, 1, 1]$. The function is modeled by the deterministic asynchronous MT-FST in Figure 1 using three input tapes: **C**-tape, **V**-tape, and **T**-tape. The transition arcs in the MT-FST in Figure are in shorthand. In a transition arc like $[c, \Sigma_{\rtimes}, C]$: $[+1, 0, +1]$: c, lower case letters are interpreted as variables. A derivation is provided in Table 2. Each row keeps track of the:

1. current state
2. location of the read heads on the 3 input tapes
3. transition arc used on each 3 input tapes
4. outputted symbol
5. current output string

We use a deterministic asynchronous MT-FST because it can *iconically* model MISL functions, while a synchronous MT-FST cannot without sacrificing locality. The reason is because synchronous MT-FSTs are equivalent to single-tape FSAs, thus making RPM computed non-locally. To illustrate, Figure 2 is the derivation for *kutib* using a synchronous 4-tape MT-FSA. To avoid asynchrony, the 3 'input' tapes are aligned with the corresponding symbols on the 'output' tape by using the special symbol □ as a padding symbol.

To understand why the function is [1,1,1]-MISL, consider its MT-FST in Figure 1. Besides the initial and final state, there is only one state q_1. q_1 keeps track of the last $\vec{k}-1$ suffix on each of the three input-strings. Because $\vec{k}-1 = [1, 1, 1]-1 =$ $[0, 0, 0]$, the state q_1 does not keep track of any previous input-symbol seen. When deciding on what to output and which state to go to, only the current input symbols on the 3 tapes were needed.

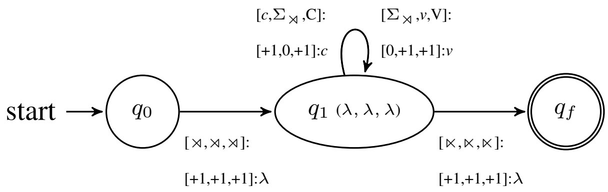


Figure 1: MT-FST for 1-to-1 slot-filling.

Input Tapes	**C**:	k	□	t	□	b
	V:	□	u	□	i	□
	T:	C	V	C	V	C
Output Tape:		k	u	t	i	b

Figure 2: Alignment of *kutib* with a synchronous MT-FSA (cf. Kiraz, 2001; Hulden, 2009).

4.1.2 1-to-1 slot-filling with four or more consonants

Extensions of 1-to-1 slot-filling are also [1,1,1]-MISL. If the root contains four consonants **C**=*trʒm* and the template has four consonant slots **T**=*CVCCVC* (Table 1b), then the output *turʒim* is generated with the same [1,1,1]-MISL function that's modeled by the MT-FST in Figure 1. A sample derivation is provided in the appendix.

If the root contains more consonants **C**=*mɣnṭs* than the template has consonant slots **T**=*CVCCVC* (Table 1c), the output shows deletion of the additional consonant: *muɣniṭ* not **muɣniṭs*. This is [1,1,1]-MISL. It is modeled by the same MT-FST in Figure 1 but with the additional transition arc: $[c, \Sigma_{\rtimes}, \ltimes]$: $[+1, 0, 0]$: λ between q_1, q_1. A sample FST and derivation are provided in the appendix.

4.1.3 1-to-1 slot-filling and pre-associated affixes

Given a root **C**=*ksb*, some outputs show an additional affix, e.g. the infix <*t*> in *k*<*t*>*usib*. The affix <t> is pre-associated to a slot after the first consonant. Pre-associated templates can be computed either representationally or derivationally. Both are local.[7]

[7] A third alternative is to treat the infix <*t*> as part of a separate input-string or input-tape. The template is *CCVCVC* where **C** is pre-associated to <*t*>. This is analogous to giving each morpheme its own autosegmental tier (McCarthy,

	Current State	C-tape		V-tape		T-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}$ktb$\ltimes$		$\underline{\rtimes}$ui$\ltimes$		$\underline{\rtimes}$CVCVC$\ltimes$			
2.	q_1	$\rtimes\underline{\text{k}}tb\ltimes$	**C**:$\rtimes$:+1	$\rtimes\underline{\text{u}}i\ltimes$	**V**:$\rtimes$:+1	$\rtimes\underline{\text{C}}$VCVC$\ltimes$	**T**:$\rtimes$:+1	λ	
3.	q_1	$\rtimes$k$\underline{\text{t}}$b$\ltimes$	**C**:*k*:+1	$\rtimes\underline{\text{u}}i\ltimes$	**V**:*u*:0	$\rtimes$C$\underline{\text{V}}$CVC$\ltimes$	**T**:*c*:+1	*k*	*k*
4.	q_1	$\rtimes$k$\underline{\text{t}}$b$\ltimes$	**C**:*t*:0	$\rtimes$u$\underline{\text{i}}\ltimes$	**V**:*u*:+1	$\rtimes$CV$\underline{\text{C}}$VC$\ltimes$	**T**:*v*:+1	*u*	*ku*
5.	q_1	$\rtimes$kt$\underline{\text{b}}\ltimes$	**C**:*t*:+1	$\rtimes$u$\underline{\text{i}}\ltimes$	**V**:*i*:0	$\rtimes$CVC$\underline{\text{V}}$C$\ltimes$	**T**:*c*:+1	*t*	*kut*
6.	q_1	$\rtimes$kt$\underline{\text{b}}\ltimes$	**C**:*b*:0	$\rtimes$ui$\underline{\ltimes}$,	**V**:*i*:+1	$\rtimes$CVCV$\underline{\text{C}}\ltimes$	**T**:*v*:+1	*i*	*kuti*
7.	q_1	$\rtimes$ktb$\underline{\ltimes}$	**C**:*b*:+1	$\rtimes$ui$\underline{\ltimes}$	**V**:$\ltimes$:0	$\rtimes$CVCVC$\underline{\ltimes}$	**T**:*c*:+1	*b*	*kutib*
8.	q_f	$\rtimes$ktb$\ltimes$	**C**:$\ltimes$:+1	$\rtimes$ui$\ltimes$	**C**:$\ltimes$:+1	$\rtimes$CVCVC$\ltimes$	**T**:$\ltimes$:+1	λ	*kutib*

Table 2: Derivation of *kutib* using the MT-FST in Figure 1.

The representational route is to enrich the template with the affix itself: **T**=*CtVCVC* (Hudson, 1986). The root and template are then combined to generate *k<t>usib*. This function is [1,1,1]-MISL. It is computed by the same MT-FST in Figure 1 but with the additional transition arc: $[\Sigma_{\rtimes}, \Sigma_{\rtimes}, t] : [0, 0, t] : \lambda$ between q_1, q_1. A sample FST and derivation are provided in the appendix.

A derivational alternative is to derive *k<t>usib* from an un-affixed base *kusib* by infixing *<t>* (McCarthy, 1993). Generating *kusib* from [*ksb, ui, CVCVC*] is [1,1,1]-MISL. Infixing *<t>* onto *kusib* is 2-ISL. The representational route can be interpreted as the composition of the derivational approach.

4.2 1-to-many slot filling

4.2.1 Final spread

Final spread in *katab* has 1-to-many slot-filling (Table 1e). The word consists of the following input strings: **C**=*ktb*, **V**=*a*, **T**=*CVCVC*. The vocalism **V** consists of only one vowel *a* because of the Obligatory Contour Principle (McCarthy, 1981). The vowel *a* undergoes final spread by being associated with multiple *V* slots in the **T**-string.

Computing final vowel spread is [1,2,1]-MISL with $k_2 = 2$ on the **V**-string, not $k_2 = 1$. Knowing to spread the final vowel requires a window of size 2 on the **V**-string. The locality window stays at 1 for the **C**,**T**-strings because they do not play a role. For illustration, we provide an MT-FST for final vowel spread in the appendix. The states keep track of the last 1-suffix on the **V**-tape and last 0-suffix on **C**,**T**-tapes. A sample FST and derivation are provided in the appendix.

1981). But computing this type of input-structure cannot be modeled in an MT-FST because MT-FSTs work over multiple linear strings, not over graphs.

Consonants can also undergo final spread: f([*sm, a, CVCVC*] = *samam* (Table 1f).[8] This is [2,1,1]-MISL, analogous to final spread of vowels except that the locality window is now larger over the **C**-string instead of the **V**-string.

4.2.2 Medial spread

In contrast to final spread, medial spread involves associating a string-medial vowel or consonant to multiple slots on the **T**-string: *kuutib* with a long-vowel *u* (Table 1g) or *kuttib* with a geminate *t* (Table 1h). Like pre-associated affixes (§4.1.3), medial spread can be analyzed either representationally or derivationally. An alternative edge-in analysis is discussed in §5.2.

For gemination, the representational route involves enriching the template with a special symbol, i.e., a consonant mora μ_C in **T**=$CVC\mu_V VC$ (Kay, 1987; McCarthy, 1993; Beesley, 1998). With this template, generating *kuttib* is [2,1,1]-MISL with k_1=2 over the **C**-string. A corresponding MT-FST and derivation is in the appendix using $\Sigma_T = \{C, V, \mu_C\}$, and $\Sigma_C = \{k, t\}$ for illustration. Long vowels have the same computational treatment but with μ_V as a special symbol.

A derivational alternative is to derive *kuttib* from *kutib* by infixing a consonant mora μ_C followed by consonant spreading. Generating the base *kutib* is [1,1,1]-MISL. Infixing the mora *kut*μ_C*ib* is 4-ISL and spreading the consonant *kuttib* is 2-ISL. As with preassociation (§4.1.3), the

[8]Since McCarthy (1981), the analysis of final consonant spread has been controversial (Hudson, 1986; Hoberman, 1988; Yip, 1988; McCarthy, 1993; Gafos, 1998; Bat-El, 2006). Alternative analyses involving reduplication, preference for local spreading, or right-to-left association can be potentially non-local and are discussed in §5. Computationally, Beesley (1998) formalizes consonant spread with a special symbol X as an equivalent treatment for medial spread. This formalization is [2,1,1]-MISL, just like (§4.2.2.

representational solution is a composition of the derivational solution; both are local functions.

5 Possible non-locality in Semitic

Certain templatic processes in Semitic are not local: reduplication and loanword adaptation in Table 3, amassed from many sources (McCarthy, 1981; Broselow and McCarthy, 1983; Bat-El, 2011).

5.1 Reduplication

Semitic RPM shows intensive reduplication which varies on root size (Broselow and McCarthy, 1983): root doubling in for biconsonantal roots in *laflaf* (Table 3i) and first-C copying for triconsonantal roots in *barbad* (Table 3j). Root-doubling is analogous to total reduplication. Initial-C copying involves copying the first consonant of the root and placing it in a prespecified spot on the template.[9]

Reduplication is computationally challenging. Cross-linguistically, partial reduplication patterns can range from being ISL to subsequential (Chandlee and Heinz, 2012). Total reduplication is above the subsequential threshold and cannot be modeled by 1-way FSTs but requires deterministic 2-way FSTs (Dolatian and Heinz, 2018). If we assume that there's no bound on the size of the root, then root-doubling cannot be computed by a MISL function for any $\vec{k}$. The function would need a 2-way MT-FST which could go back and forth on the **C**-tape. Similarly, if we assume that there's no bound on the number n of consonants between the two copies of the root-initial consonant, then the function is not MISL for any $\vec{k}$. Analogously to subsequential functions over single-input FSTs, root-initial copying would be Multi-Subsequential. However, the assumption on root size is not correct. All roots which undergo the above reduplication processes have a bounded size (2 or 3). If we discard this assumption, then both reduplicative processes are MISL for a large value of $\vec{k}$.[10]

5.2 Local spreading in loanword adaptation

In loanword adaptation of verbs in Arabic, the most productive template is *CVCCVC* with the vocalism *a*: *CaCCaC* (Bat-El, 2011). When a borrowed consonantal root has four consonants, the template is filled with 1-to-1 slot filling of consonants: *telephone* [telefon] and *talfan* (Table 3k). But when a borrowed root has three consonants, then the input undergoes medial gemination: *SMS* and *sammas*, not final spread **samsas* (Table 3l).

There are many ways to analyze this difference between three vs. four-consonant roots. One is suppletive allomorphy: four-consonant roots use the template *CVCCVC*, three-consonant roots use the template $CVC\mu_C VC$. Choosing the template is ISL-4. Once chosen, the root, vocalism, and template can then be submitted to an MISL function. This analysis is plausible because, outside of loanword adaptation, Semitic templates *do* have suppletion conditioned by root-size: the comparative in Egyptian Arabic is *VCCVC* for triconsonantal roots: *kbr* → *akbar*, but *VCVCC* for biconsonantal roots: *ʃd* → *aʃadd* (Davis and Tsujimura, 2018).

An alternative is to use a template *CVC-CVC* without any representational markup for gemination. The correct outputs are generated based on avoiding non-local spreading. For a three-consonant root, medial gemination is generated because the grammar (in OT-parlance) prefers outputs with local spreading of consonants *sa***mm***as* instead of outputs with non-local spreading *sam***s***as*. An analogous anti-long-distance spreading mechanism has been proposed for medial gemination (§4.2.2) and for the fact that *i* cannot spread (§4.2.1) (Hudson, 1986; Hoberman, 1988; Yip, 1988).[11] Computationally, the choice of local spreading depends on the following information:

1. Having the context CCV on the template: $k = 3$ on **T**-string
2. Being the final consonant in the root or not: $k = 2$ on **C**-string
3. The existence of an additional C slot on the template: $XCCV_yC⋉$ vs. $\mathrm{XCCV_y}⋉$: $k = |V_\mathrm{x}| + 1$ on **T**-string

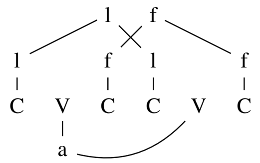
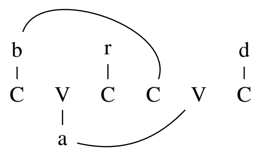
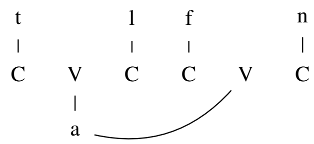
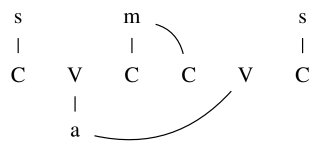
The last condition is important. Consider the contrast in *kuttib* and *kutba* 'writers' derived from the templates $C_1V_xC_2C_3V_yC_4$ and $C_1V_xC_2C_3V_y$.

[9] Technically, the relevant inputs need to be annotated to trigger reduplication, e.g. initial-C copying with **T**=*CVCFVC* and root doubling with **C**=*zl*-RED. We abstract away from this for clarity.

[10] The value of the k is [3,1,1] for initial-C copying, but [3,1,3] for root-doubling because the function keeps track of the root size and the current C-slot.

[11] These have also been analyzed with edge-in association. Instead of association operating from left-to-right, Yip (1988) argues that these templates are simultaneously or consecutively right-to-left and left-to-right. Such an analysis though has unclear computational expressivity; we conjecture that it may be analogous to Weak Determinism (Heinz and Lai, 2013) over multiple inputs.

Table 3: Partial paradigm of Arabic root-and-pattern morphology with variable MISL $\vec{k}$-values.

	Slot-filling pattern	Binyan	Gloss	Output	Root	Vowels	Template	k-value
	Reduplication of							
i	... root		*laflaf*	'wrapped intensely'	*lf*	*a*	*CVCCVC* 	varies
j	... first C		*barbad*	'shaved unevenly'	*brd*	*a*	*CVCFVC* 	varies
	Loanword adaptation of...	Source noun	Adapted Verb					
k	... four-consonant root	*telephone*	*talfan*	'he phoned'	*tlfn*	*a*	*CVCCVC* 	varies
l	... three-consonant root	*SMS*	*sammas*	'he SMS-ed'	*sms*	*a*	*CVCCVC* 	varies

The C_2C_3 substring in $C_1V_xC_2C_3V_yC_4$ maps to gemination: *kuttib*, while the *CC* substring in *CVCCV* maps to 1-to-1 spreading: *kutba*. The choice depends on if the C_1C_2 substrings precedes an extra consonant slot C_4 on the template or not. If there is no bound on the number of intervening vowels V_x, then the function is not MISL for any $\vec{k}$. If there is a bound, then it is MISL for a k which is sufficiently large enough to encode these contexts. In Arabic, V_y can be at most two vowels slots in order to encode long vowels: *kuttaab* 'writers'. This makes the function MISL with $k = 5$ on the **T**-string, $k = 3$ on the **C**-string.

6 Conclusion

This paper examined the computational expressivity of non-concatenative morphology, in particular, Semitic root-and-pattern morphology (RPM). Generalizing Input Strictly Local (ISL) functions to handle multiple inputs, we showed that the class of Multiple-Input Strictly Local (MISL) functions can compute almost all Semitic RPM. These MISL functions are computed by deterministic asynchronous multi-tape finite-state transducers. This computational result looks beyond various points of theoretical contention in Semitic. The result also narrows the gap in mathematical results between concatenative and non-concatenative morphology.

References

Mark Aronoff. 2013. The roots of language. In Silvio Cruschina, Martin Maiden, , and John Charles Smith, editors, The boundaries of pure morphology, pages 161–180.

Outi Bat-El. 2006. Consonant identity and consonant copy: The segmental and prosodic structure of hebrew reduplication. Linguistic Inquiry, 37(2):179–210.

Outi Bat-El. 2011. Semitic templates. In (van Oostendorp et al., 2011), pages 2586–2609.

Kenneth Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. CSLI Publications, Stanford, CA.

Kenneth R Beesley. 1998. Consonant spreading in arabic stems. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on

Computational Linguistics-Volume 1, pages 117–123. Association for Computational Linguistics.

Kenneth R. Beesley and Lauri Karttunen. 2000. Finite-state non-concatenative morphotactics. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, pages 191–198, Hong Kong. Association for Computational Linguistics.

Steven Bird and T Mark Ellison. 1994. One-level phonology: Autosegmental representations and rules as finite automata. Computational Linguistics, 20(1):55–90.

Ellen Broselow and John McCarthy. 1983. A theory of internal reduplication. The Linguistic Review, 3(1):25–88.

Jane Chandlee. 2014. Strictly Local Phonological Processes. Ph.D. thesis, University of Delaware, Newark, DE.

Jane Chandlee. 2017. Computational locality in morphological maps. Morphology, pages 1–43.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015a. Output strictly local functions. In 14th Meeting on the Mathematics of Language, pages 112–125.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015b. Output strictly local functions. In Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), pages 112–125, Chicago, USA.

Jane Chandlee and Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, SIGMORPHON '12, pages 42–51, Montreal, Canada. Association for Computational Linguistics.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. Linguistic Inquiry, 49(1):23–60.

Yael Cohen-Sygal and Shuly Wintner. 2006. Finite-state registered automata for non-concatenative morphology. Computational Linguistics, 32(1):49–82.

Stuart Davis and Natsuko Tsujimura. 2018. Arabic nonconcatenative morphology in construction morphology. In Geert Booij, editor, The Construction of Words: Advances in Construction Morphology, volume 4. Springer.

Hossep Dolatian and Jeffrey Heinz. 2018. Modeling reduplication with 2-way finite-state transducers. In Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Brussells, Belgium. Association for Computational Linguistics.

Hossep Dolatian and Jonathan Rawski. 2019. Finite-state locality in semitic root-and-pattern morphology. In University of Pennsylvania Working Papers in Linguistics.

C. C. Elgot and J. E. Mezei. 1965. On relations defined by generalized finite automata. IBM Journal of Research and Development, 9(1):47–68.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. ACM SIGLOG News, 3(3):4–19.

Patrick C Fischer. 1965. Multi-tape and infinite-state automataa survey. Communications of the ACM, 8(12):799–805.

Patrick C Fischer and Arnold L Rosenberg. 1968. Multitape one-way nonwriting automata. Journal of Computer and System Sciences, 2(1):88–101.

Christiane Frougny and Jacques Sakarovitch. 1993. Synchronized rational relations of finite and infinite words. Theoretical Computer Science, 108(1):45–82.

Carlo A. Furia. 2012. A survey of multi-tape automata. http://arxiv.org/abs/1205.0178. Latest revision: November 2013.

Diamandis Gafos. 1998. Eliminating long-distance consonantal spreading. Natural Language & Linguistic Theory, 16(2):223–278.

Michael Hammond. 1988. Templatic transfer in arabic broken plurals. Natural Language & Linguistic Theory, 6(2):247–270.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13), pages 52–63, Sofia, Bulgaria. Association for Computational Linguistics.

Robert D Hoberman. 1988. Local and long-distance spreading in semitic morphology. Natural Language & Linguistic Theory, 6(4):541–549.

Grover Hudson. 1986. Arabic root and pattern morphology without tiers. Journal of Linguistics, 22(1):85–122.

Mans Hulden. 2009. Revisiting multi-tape automata for semitic morphological analysis and generation. In Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, pages 19–26. Association for Computational Linguistics.

Itamar Kastner. 2016. Form and meaning in the Hebrew verb. Ph.D. thesis, New York University.

Martin Kay. 1987. Nonconcatenative finite-state morphology. In Third Conference of the European Chapter of the Association for Computational Linguistics.

George Anton Kiraz. 2000. Multitiered nonlinear morphology using multitape finite automata: a case study on syriac and arabic. Computational Linguistics, 26(1):77–105.

George Anton Kiraz. 2001. Computational nonlinear morphology: with emphasis on Semitic languages. Cambridge University Press.

John McCarthy and Alan Prince. 1990a. Prosodic morphology and templatic morphology. In Perspectives on Arabic linguistics II: papers from the second annual symposium on Arabic linguistics, pages 1–54. John Benjamins Amsterdam.

John J McCarthy. 1981. A prosodic theory of nonconcatenative morphology. Linguistic inquiry, 12(3):373–418.

John J McCarthy. 1993. Template form in prosodic morphology. In Proceedings of the Formal Linguistics Society of Mid-America, volume 3, pages 187–218.

John J McCarthy and Alan S Prince. 1990b. Foot and word in prosodic morphology: The Arabic broken plural. Natural Language & Linguistic Theory, 8(2):209–283.

Robert McNaughton and Seymour A Papert. 1971. Counter-Free Automata (MIT research monograph no. 65). The MIT Press.

Marc van Oostendorp, Colin Ewen, Elizabeth Hume, and Keren Rice, editors. 2011. The Blackwell companion to phonology. Wiley-Blackwell, Malden, MA.

Jean-François Prunet. 2006. External evidence and the semitic root. Morphology, 16(1):41.

Michael O Rabin and Dana Scott. 1959. Finite automata and their decision problems. IBM journal of research and development, 3(2):114–125.

Brian Roark and Richard Sproat. 2007. Computational Approaches to Morphology and Syntax. Oxford University Press, Oxford.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In Formal Grammar, volume 8036 of Lecture Notes in Computer Science, pages 90–108. Springer.

James Rogers and Geoffrey Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. Journal of Logic, Language and Information, 20:329–342.

Walter J Savitch. 1993. Why it might pay to assume that languages are infinite. Annals of Mathematics and Artificial Intelligence, 8(1-2):17–25.

Matthew A Tucker. 2010. Roots and prosody: the iraqi arabic derivational verb. Recherches linguistiques de Vincennes, (39):31–68.

Adam Ussishkin. 2005. A fixed prosodic theory of nonconcatenative templaticmorphology. Natural Language & Linguistic Theory, 23(1):169–218.

Adam Ussishkin. 2011. Tier segregation. In (van Oostendorp et al., 2011), pages 2516–2535.

Bruce Wiebe. 1992. Modelling autosegmental phonology with multi-tape finite state transducers. Master's thesis, Simon Fraser University.

Shuly Wintner. 2014. Morphological processing of semitic languages. In Imed Zitouni, editor, Natural language processing of Semitic languages, pages 43–66. Springer.

Moira Yip. 1988. Template morphology and the direction of association. Natural Language & Linguistic Theory, 6(4):551–577.

Sam Zukoff. 2017. Arabic nonconcatenative morphology and the syntax-phonology interface. In NELS 47: Proceedings of the Forty-Seventh Annual Meeting of the North East Linguistic Society, volume 3, page 295314, Amherst, MA. Graduate Linguistics Student Association.

A Appendix

Below are MT-FSTs and derivation tables for some of the described Semitic processes.

A.1 1-to-1 slot-filling with four consonants

In Table 1b, the input root **C** has 4 consonants *trʒm* and the template **T** has enough consonantal slots *CVCCVC*. The vocalism **V** is *ui*. The output is *turʒim*. A derivation table is provided in Table 4 using the [1,1,1]-MISL MT-FST from Figure 1.

A.2 1-to-1 slot-filling with larger roots

In Table 1c, the root **C**==*mɣnts* contains more consonants than the template **T**=*CVCCVC*. With a vocalism **V**=*ui*, the output is *muɣnit* with final consonant deletion. This function is modeled by the [1,1,1]-MISL MT-FST in Figure 3, illustrated with the derivation in Table 5.

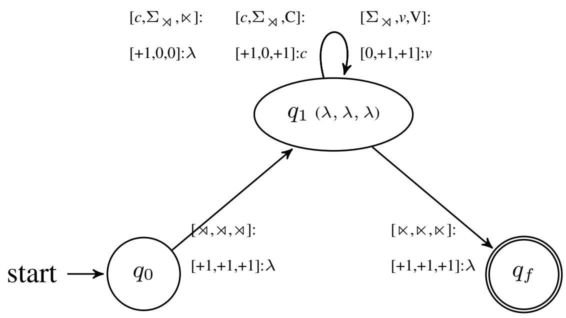


Figure 3: MT-FST for 1-to-1 slot-filling with final consonant deletion

A.3 1-to-1 slot-filling and pre-associated affixes

The template **T**=*CtVCVC* has a preassociated affix $\langle t \rangle$. With a root **C**=*ksb* and vocalism **V**=*ui*, the output is *ktusib*. A [1,1,1]-MISL MT-FST is provided in Figure 4 along with a sample derivation in Table 6. The symbol *A* represents any input symbol from the input alphabet of segments {t,n,m} which are possible segmental affixes in McCarthy (1981).

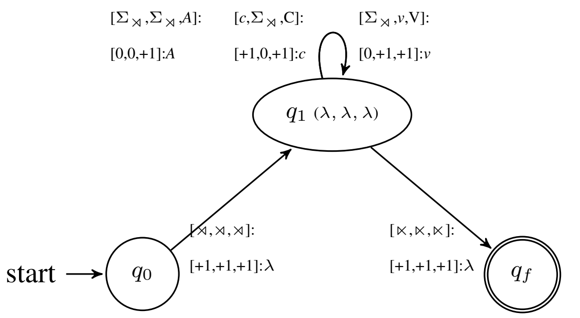


Figure 4: MT-FST for 1-to-1 slot-filling with pre-associated affixes

A.4 1-to-many slot-filling with final spread of vowels

In Table 1e, the vocalism **V**=*a* has fewer vowels than the template **T**=*CVCVC*. This triggers final spread of vowels. With a root **C**=*ktb*, the output is *katab*. This function is modeled with the [1,2,1]-MISL MT-FST in Figure 5, illustrated with a sample derivation in Table 7. The vowel alphabet is only {a,u}. In Standard Arabic, only the vowels *a,u* spread; the vowel *i* does not. This is discussed in §5.2. The FST does not visually represent the dedicated final state q_f. Instead, all non-initial states are marked as accepting states. A state is accepting if upon reading [⋉,⋉,⋉], it advances [+1,+1,+1] to state q_f.

A.5 1-to-many slot filling with medial spread of consonants

In Table 1g, the template **T**=*CVC*μ_C*VC* contains a marker for gemination. With root **C**=*ktb* and vocalism **V**=*ui*, the output is *kuttib*. This is modeled by the [2,1,1]-MISL MT-FST in Figure 6. with a sample derivation in Table 8 for a nonce word *kuttik* with root **C**=*ktk*. For illustrative reasons, the consonant alphabet is only {k,t}. The final state q_f is not visualized for space reasons.

	Current State	**C**-tape		**V**-tape		**T**-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}\text{trʒm}\ltimes$		$\underline{\rtimes}\text{ui}\ltimes$		$\underline{\rtimes}\text{CVCCVC}\ltimes$			
2.	q_1	$\rtimes\underline{\text{t}}\text{rʒm}\ltimes$	$\mathbf{C}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{C}}\text{VCCVC}\ltimes$	$\mathbf{T}{:}\rtimes{:}{+1}$	λ	
3.	q_1	$\rtimes\text{t}\underline{\text{r}}\text{ʒm}\ltimes$	$\mathbf{C}{:}t{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}u{:}0$	$\rtimes\text{C}\underline{\text{V}}\text{CCVC}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*t*	*t*
4.	q_1	$\rtimes\text{t}\underline{\text{r}}\text{ʒm}\ltimes$	$\mathbf{C}{:}r{:}0$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}u{:}{+1}$	$\rtimes\text{CV}\underline{\text{C}}\text{CVC}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*u*	*tu*
5.	q_1	$\rtimes\text{tr}\underline{\text{ʒ}}\text{m}\ltimes$	$\mathbf{C}{:}r{:}{+1}$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}i{:}0$	$\rtimes\text{CVC}\underline{\text{C}}\text{VC}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*r*	*tur*
6.	q_1	$\rtimes\text{trʒ}\underline{\text{m}}\ltimes$	$\mathbf{C}{:}ʒ{:}{+1}$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}i{:}0$	$\rtimes\text{CVCC}\underline{\text{V}}\text{C}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*ʒ*	*turʒ*
7.	q_1	$\rtimes\text{trʒ}\underline{\text{m}}\ltimes$	$\mathbf{C}{:}m{:}0$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}i{:}{+1}$	$\rtimes\text{CVCCV}\underline{\text{C}}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*i*	*turʒi*
8.	q_1	$\rtimes\text{trʒm}\underline{\ltimes}$	$\mathbf{C}{:}m{:}{+1}$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}\ltimes{:}0$	$\rtimes\text{CVCCVC}\underline{\ltimes}$	$\mathbf{T}{:}C{:}{+1}$	*m*	*turʒim*
9.	q_1	$\rtimes\text{trʒm}\ltimes$	$\mathbf{C}{:}\ltimes{:}{+1}$	$\rtimes\text{ui}\ltimes$	$\mathbf{V}{:}\ltimes{:}{+1}$	$\rtimes\text{CVCCVC}\ltimes$	$\mathbf{T}{:}\ltimes{:}{+1}$	λ	*turʒim*

Table 4: Derivation of *turʒim* using the MT-FST in Figure 1.

	Current State	**C**-tape		**V**-tape		**T**-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}\text{mɣnts}\ltimes$		$\underline{\rtimes}\text{ui}\ltimes$		$\underline{\rtimes}\text{CVCCVC}\ltimes$			
2.	q_1	$\rtimes\underline{\text{m}}\text{ɣnts}\ltimes$	$\mathbf{C}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{C}}\text{VCCVC}\ltimes$	$\mathbf{T}{:}\rtimes{:}{+1}$	λ	
3.	q_1	$\rtimes\text{m}\underline{\text{ɣ}}\text{nts}\ltimes$	$\mathbf{C}{:}m{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}u{:}0$	$\rtimes\text{C}\underline{\text{V}}\text{CCVC}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*m*	*m*
4.	q_1	$\rtimes\text{m}\underline{\text{ɣ}}\text{nts}\ltimes$	$\mathbf{C}{:}ɣ{:}0$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}u{:}{+1}$	$\rtimes\text{CV}\underline{\text{C}}\text{CVC}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*u*	*mu*
5.	q_1	$\rtimes\text{mɣ}\underline{\text{n}}\text{ts}\ltimes$	$\mathbf{C}{:}ɣ{:}{+1}$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}i{:}0$	$\rtimes\text{CVC}\underline{\text{C}}\text{VC}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*G*	*muG*
6.	q_1	$\rtimes\text{mɣn}\underline{\text{t}}\text{s}\ltimes$	$\mathbf{C}{:}n{:}{+1}$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}i{:}0$	$\rtimes\text{CVCC}\underline{\text{V}}\text{C}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*n*	*muGn*
7.	q_1	$\rtimes\text{mɣn}\underline{\text{t}}\text{s}\ltimes$	$\mathbf{C}{:}t{:}0$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}i{:}{+1}$	$\rtimes\text{CVCCV}\underline{\text{C}}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*i*	*muGni*
8.	q_1	$\rtimes\text{mɣnt}\underline{\text{s}}\ltimes$	$\mathbf{C}{:}t{:}{+1}$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}\ltimes{:}0$	$\rtimes\text{CVCCVC}\underline{\ltimes}$	$\mathbf{T}{:}C{:}{+1}$	*t*	*muGnit*
9.	q_1	$\rtimes\text{mɣnts}\underline{\ltimes}$	$\mathbf{C}{:}s{:}{+1}$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}\ltimes{:}0$	$\rtimes\text{CVCCVC}\underline{\ltimes}$	$\mathbf{T}{:}\ltimes{:}0$	λ	*muGnit*
10.	q_1	$\rtimes\text{mɣnts}\ltimes$	$\mathbf{C}{:}\ltimes{:}{+1}$	$\rtimes\text{ui}\ltimes$	$\mathbf{V}{:}\ltimes{:}{+1}$	$\rtimes\text{CVCCVC}\ltimes$	$\mathbf{T}{:}\ltimes{:}{+1}$	λ	*muGnit*

Table 5: Derivation of *muɣnit* using the MT-FST in Figure 3

	Current State	**C**-tape		**V**-tape		**T**-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}\text{ksb}\ltimes$		$\underline{\rtimes}\text{ui}\ltimes$		$\underline{\rtimes}\text{CtVCVC}\ltimes$			
2.	q_1	$\rtimes\underline{\text{k}}\text{sb}\ltimes$	$\mathbf{C}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}\rtimes{:}{+1}$	$\rtimes\underline{\text{C}}\text{tVCVC}\ltimes$	$\mathbf{T}{:}\rtimes{:}{+1}$	λ	
3.	q_1	$\rtimes\text{k}\underline{\text{s}}\text{b}\ltimes$	$\mathbf{C}{:}k{:}{+1}$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}u{:}0$	$\rtimes\text{C}\underline{\text{t}}\text{VCVC}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*k*	*k*
4.	q_1	$\rtimes\text{k}\underline{\text{s}}\text{b}\ltimes$	$\mathbf{C}{:}s{:}0$	$\rtimes\underline{\text{u}}\text{i}\ltimes$	$\mathbf{V}{:}u{:}0$	$\rtimes\text{Ct}\underline{\text{V}}\text{CVC}\ltimes$	$\mathbf{T}{:}t{:}{+1}$	*t*	*kt*
5.	q_1	$\rtimes\text{k}\underline{\text{s}}\text{b}\ltimes$	$\mathbf{C}{:}s{:}0$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}u{:}{+1}$	$\rtimes\text{CtV}\underline{\text{C}}\text{VC}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*u*	*ktu*
6.	q_1	$\rtimes\text{ks}\underline{\text{b}}\ltimes$	$\mathbf{C}{:}s{:}{+1}$	$\rtimes\text{u}\underline{\text{i}}\ltimes$	$\mathbf{V}{:}i{:}0$	$\rtimes\text{CtVC}\underline{\text{V}}\text{C}\ltimes$	$\mathbf{T}{:}C{:}{+1}$	*s*	*ktus*
7.	q_1	$\rtimes\text{ks}\underline{\text{b}}\ltimes$	$\mathbf{C}{:}b{:}0$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}i{:}{+1}$	$\rtimes\text{CtVCV}\underline{\text{C}}\ltimes$	$\mathbf{T}{:}V{:}{+1}$	*i*	*ktusi*
8.	q_1	$\rtimes\text{ksb}\underline{\ltimes}$	$\mathbf{C}{:}b{:}{+1}$	$\rtimes\text{ui}\underline{\ltimes}$	$\mathbf{V}{:}\ltimes{:}0$	$\rtimes\text{CtVCVC}\underline{\ltimes}$	$\mathbf{T}{:}C{:}{+1}$	*b*	*ktusib*
9.	q_1	$\rtimes\text{ksb}\ltimes$	$\mathbf{C}{:}\ltimes{:}{+1}$	$\rtimes\text{ui}\ltimes$	$\mathbf{V}{:}\ltimes{:}{+1}$	$\rtimes\text{CtVCVC}\ltimes$	$\mathbf{T}{:}\ltimes{:}{+1}$	λ	*ktusib*

Table 6: Derivation of $k\langle t\rangle$*usib* using the MT-FST in Figure 4

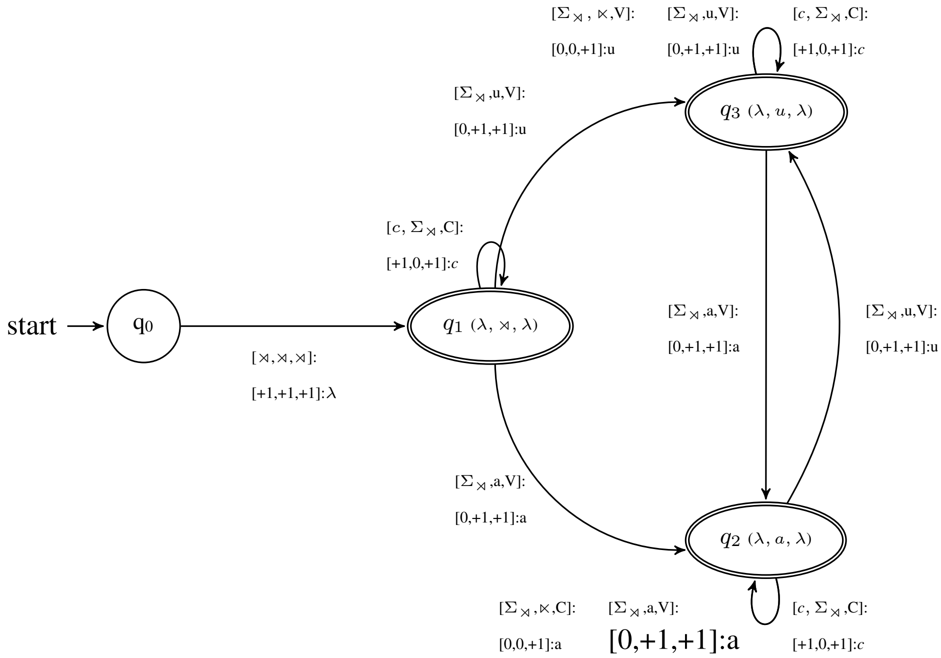


Figure 5: MT-FST for 1-to-many slot-filling with final spread of vowels

	Current State	C-tape		V-tape		T-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}$ktb$\ltimes$		$\underline{\rtimes}$a$\ltimes$		$\underline{\rtimes}$CVCVC$\ltimes$			
2.	q_1	$\rtimes\underline{k}$tb$\ltimes$	**C**:$\rtimes$:+1	$\rtimes\underline{a}\ltimes$	**V**:$\rtimes$:+1	$\rtimes\underline{C}$VCVC$\ltimes$	**T**:$\rtimes$:+1	λ	
3.	q_1	$\rtimes$k$\underline{t}$b$\ltimes$	**C**:k:+1	$\rtimes\underline{a}\ltimes$	**V**:a:0	$\rtimes$C$\underline{V}$CVC$\ltimes$	**T**:C:+1	*k*	*k*
4.	q_2	$\rtimes$k$\underline{t}$b$\ltimes$	**C**:t:0	$\rtimes$a$\underline{\ltimes}$	**V**:a:+1	$\rtimes$CV$\underline{C}$VC$\ltimes$	**T**:V:+1	*a*	*ka*
5.	q_2	$\rtimes$kt$\underline{b}\ltimes$	**C**:t:+1	$\rtimes$a$\underline{\ltimes}$	**V**:$\ltimes$:0	$\rtimes$CVC$\underline{V}$C$\ltimes$	**T**:t:+1	*t*	*kat*
6.	q_2	$\rtimes$kt$\underline{b}\ltimes$	**C**:b:0	$\rtimes$a$\underline{\ltimes}$,	**V**:$\ltimes$:0	$\rtimes$CVCV$\underline{C}\ltimes$	**T**:V:+1	*a*	*kata*
7.	q_2	$\rtimes$ktb$\underline{\ltimes}$	**C**:b:+1	$\rtimes$a$\underline{\ltimes}$	**V**:$\ltimes$:0	$\rtimes$CVCVC$\underline{\ltimes}$	**T**:C:+1	*b*	*katab*
8.	q_f	$\rtimes$ktb$\ltimes$	**C**:$\ltimes$:+1	$\rtimes$a$\ltimes$	**C**:$\ltimes$:+1	$\rtimes$CVCVC$\ltimes$	**T**:$\ltimes$:+1	λ	*katab*

Table 7: Derivation of *katab* using the MT-FST in Figure 5

	Current State	C-tape		V-tape		T-tape		Output Symbol	Output String
1.	q_0	$\underline{\rtimes}$ktk$\ltimes$		$\underline{\rtimes}$ui$\ltimes$		$\underline{\rtimes}$CVCμ_CVC$\ltimes$			
2.	q_1	$\rtimes\underline{k}$tk$\ltimes$	**C**:$\rtimes$:+1	$\rtimes\underline{u}$i$\ltimes$	**V**:$\rtimes$:+1	$\rtimes\underline{C}$VCμ_CVC$\ltimes$	**T**:$\rtimes$:+1	λ	
3.	q_2	$\rtimes$k$\underline{t}$k$\ltimes$	**C**:k:+1	$\rtimes\underline{u}$i$\ltimes$	**V**:u:0	$\rtimes$C$\underline{V}$Cμ_CVC$\ltimes$	**T**:C:+1	*k*	*k*
4.	q_2	$\rtimes$k$\underline{t}$k$\ltimes$	**C**:k:0	$\rtimes$u$\underline{i}\ltimes$	**V**:u:+1	$\rtimes$CV$\underline{C}\mu_C$VC$\ltimes$	**T**:V:+1	*u*	*ku*
5.	q_3	$\rtimes$kt$\underline{k}\ltimes$	**C**:t:+1	$\rtimes$u$\underline{i}\ltimes$	**V**:i:0	$\rtimes$CVC$\underline{\mu_C}$VC$\ltimes$	**T**:C:+1	*t*	*kut*
6.	q_3	$\rtimes$kt$\underline{k}\ltimes$	**C**:k:0	$\rtimes$u$\underline{i}\ltimes$	**V**:i:0	$\rtimes$CVC$\mu_C\underline{V}$C$\ltimes$	**T**:μ_C:+1	*t*	*kutt*
7.	q_3	$\rtimes$kt$\underline{k}\ltimes$	**C**:k:0	$\rtimes$ui$\underline{\ltimes}$	**V**:i:+1	$\rtimes$CVCμ_CV$\underline{C}\ltimes$	**T**:V:+1	*i*	*kutti*
8.	q_3	$\rtimes$ktk$\underline{\ltimes}$	**C**:k:+1	$\rtimes$ui$\underline{\ltimes}$	**V**:$\ltimes$:0	$\rtimes$CVCμ_CVC$\underline{\ltimes}$	**T**:C:+1	*k*	*kuttik*
9.	q_f	$\rtimes$ktk$\ltimes$	**C**:$\ltimes$:+1	$\rtimes$ui$\ltimes$	**V**:$\ltimes$:+1	$\rtimes$CVCμ_CVC$\ltimes$	**T**:$\ltimes$:+1	λ	*kuttik*

Table 8: Derivation of *kuttik* using the MT-FST in Figure 6

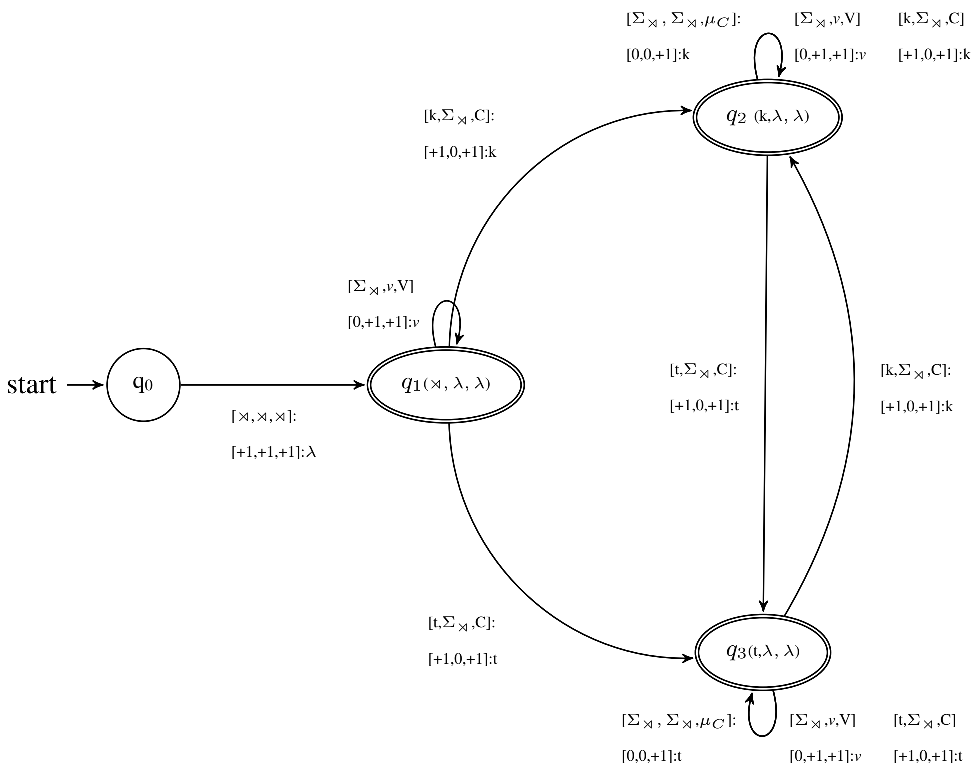


Figure 6: MT-FST for 1-to-many slot-filling with medial spread of consonants

Lexical databases for computational analyses: A linguistic perspective

Robert Malouf[1], **Farrell Ackerman**[2], **and Arturs Semenuks**[3]
[1]Department of Linguistics and Asian/Middle Eastern Languages, San Diego State University
`rmalouf@sdsu.edu`
[2]Department of Linguistics, University of California, San Diego
`fackerman@ucsd.edu`
[3]Department of Cognitive Science, University of California, San Diego
`asemenuk@ucsd.edu`

Abstract

Large typological databases have permitted new ways of studying cross-linguistic morphological variation. Recently, computational modelers with typological interests have begun to turn to broad multilingual text databases. In this paper, we will focus particularly on the UniMorph database, a collection of morphological paradigms, mostly gathered automatically from the crowd-sourced multi-lingual dictionary Wiktionary. It was designed to make the large quantity of data contained in Wiktionary available for NLP researchers by standardizing the data and putting it into a form that is easy to access. For typological studies, however, the requirements for a linguistically informed view of morphological variation are quite different. They involve using a morphological database as a scientific instrument to both formulate and test hypotheses about the nature and organization of language systems. The requirements are, accordingly, much higher. In this paper, we survey some of the methodological challenges and pitfalls involved in using corpora for typological research, and we end with a proposal for best practices and directions for further research.

1 Introduction

The availability of large typological databases (e.g., Dryer and Haspelmath 2013; Bickel and Nichols 2002) has made it possible to both model and hypothesize about the nature of cross-linguistic morphological variation. Recently, computational modelers with typological interests have begun to turn to broad multilingual text databases (e.g., Key and Comrie 2015; Dellert and Jäger 2017; McCarthy et al. 2018). While working from raw linguistic data opens up the possibility for new kinds of discoveries, it also poses significant challenges for the analyst, both with respect to the appropriateness of the selected data for explicitly specified goals and for identifying how these goals relate to alternatives that appeal to similar sorts of data.

Since Greenberg's (1963) pioneering work, we can roughly divide research in morphological typology into three strands. The first, and (arguably) most productive so far, has involved the careful construction of language samples designed by the author(s) of the study for answering specific questions. For example, Baerman et al. (2002, 2005) provide a cross-linguistic study of patterns of syncretism based on a database of all syncretic forms found in 30 genetically diverse languages and a larger database of person syncretisms in 111 languages, and Cysouw (2003) used a database of 102 types of person-marking system found in 309 languages.

This methodology has the advantage that both sample selection and coding is controlled by the researcher and can be designed specifically for the task at hand. However, while a few of the database entries may be based on the typologists' personal linguistic knowledge, for the most part information in the database is derived from dictionaries and grammatical descriptions, which necessarily reflect the analytic choices made by other linguists.

The second strand of typological research leverages the effort put into creating more general-purpose typological databases crafted to address multiple questions, but adaptable to address unanticipated and novel issues. For example, Bentz and Winter (2013) use the information about the case inventories of 261 languages in Iggesen (2013), which in turn is derived from Iggesen's (2005) detailed cross-linguistic study of case marking. Using existing resources in this way allows hypotheses about correlations among typolgical variables to be tested relatively easily, without months or years of work collecting language data. However, it is necessarily limited in the kinds of phenomena that can be examined, and is self-evidently dependent on the analytic choices made by the typologist who assembled the database and the linguists who wrote the grammars that the entries are based on.

Finally, a recent and very promising direction for morphological typology is the direct use of lexicons and corpora to extract cross-linguistics patterns (e.g., Wälchli and Cysouw 2012; Levshina 2016). This 'primary-data typology' has been made possible by the availability of large quantities of text in a diverse range of languages coupled with powerful statistical and computational methods. These methods allow us to investigate typological questions that cannot be addressed via grammatical descriptions. And, while all linguistic data is dependent (explicitly or implicitly) on an underlying analysis, working directly with texts makes us less dependent on the analytic choices made by other linguists. However, just as the other methodologies discussed above, this strand of typological research poses some significant challenges that researchers need to recognize and develop strategies to address.

In this paper, we will focus particularly on the UniMorph database (Kirov et al., 2016, 2018) and use it as a case study to highlight what types of obstacles 'primary-data typology' needs to take into account. UniMorph is a collection of morphological paradigms, mostly collected automatically from the crowd-sourced multi-lingual dictionary Wiktionary (`wiktionary.org`). It was designed to make the large quantity of data contained in Wiktionary available for NLP researchers by standardizing the data and encoding it in a form that is easy to access.

UniMorph has been broadly adapted as a test-bed for evaluating morphological processors (e.g., Aharoni and Goldberg 2017; Shearing et al. 2018). Its main advantage is that it is larger and simpler to use than any existing competitors. While it is plausibly preferable to use broader typological samples as a measure of progress, one can make the argument that, all databases are flawed in some way, and evaluating systems on a variety of languages, however restricted, is certainly preferable to testing on only English data. There is a danger of 'overfitting' to standard datasets as a research community, but this can be minimized by continuing to expand and improve available test sets (Kyle Gorman and Markowska, 2019).

Another promising use for resources like UniMorph is for evaluating claims about morphological systems in general separate from the tools we use to process them. For example, a number of recent papers (e.g., Cotterell et al. 2019; Pimentel et al. 2019; Wu et al. 2019) have used UniMorph to offer answers to some basic questions about the structure of morphological systems. But, in contrast to the the engineering applications of UniMorph, the requirements for engaging in such a linguistically informed view of morphological variation are quite different. They involve using a morphological database as a scientific instrument to both formulate and test hypotheses about the nature and organization of language systems. The requirements (and the stakes) are, accordingly, much higher. In linguistics, as in any other field, analysis of an inappropriate data sample can lead to misplaced confidence in unsupported conclusions and unlicensed general inferences about e.g., morphological organization.

It seems likely that the UniMorph project can form the basis of a database suitable for use in typological research, if suitably modified. Forms in the UniMorph database are annotated with features from the UniMorph Schema (Sylak-Glassman, 2016), and considerable effort was put into designing these feature representations to allow cross-linguistic comparison of categories. But, in contrast to this care, the selection of languages in the sample was made opportunistically determined by what was available in Wiktionary, rather than being selected to explore different strategies of morphological organization and related questions concerning the learnability of attested systems. These are core linguistic concerns in relation to the typological sampling of empirical phenomena.

In the following sections, we will survey some of the methodological challenges and pitfalls involved in using corpora for typological generalizing, and we will end with a proposal for best practices and directions for further research.

2 Representativeness

Any database that purports to develop generalizations about language in general has to be representative of the range of possible human languages. UniMorph[1] includes data from 106 languages, including noun paradigms for 74 and verb paradigms

[1]The version of UniMorph we use for this paper consists of all repos with three letter names containing a datafile with a three letter name in the `https://github.com/unimorph` organization, downloaded on 27 July 2019.

for 87. These languages represent 16 families (e.g., Indo-European, Uralic) and 30 genera (e.g., Celtic, Finnic). This is a very small fraction of the world's languages. By comparison, the World Atlas of Linguistic Structures (Dryer and Haspelmath, 2013) includes data for 2,679 languages representing 256 families and 544 genera in total. Or, since WALS does not include values for every feature for every language, the median feature in WALS is specified for 257 languages in 96 families and 177.5 genera.

A small sample, correctly constructed, can support cross-linquistic inferences. However, the languages in UniMorph are not representative of the diversity of human language. Almost half (47 out of 106) of the languages in UniMorph are from just three genera (Romance, Germanic, and Slavic). While the problem of individuating and enumerating languages is a difficult one with no clear solution, some of the 'languages' in UniMorph are arguably not different languages and would normally be considered dialects of a common language (e.g., German, Low German, Middle High German, and Middle Low German). Sometimes the same language is given different names and treated as if it were multiple languages for political or historical reasons.

In addition, 98 of the languages in UniMorph are spoken in Eurasia (i.e., the landmass comprising Europe and Asia) with only three languages in North America, two languages in each of South America and Africa, and only one language in Australia (see Figure 1). As Dryer (1989) demonstrated, Eurasian languages are not generally representative of languages throughout the world. This reinforces the observation that any representative sample needs to include languages with wide geographic and phylogenetic dispersion.

In addition to genetic and geographic homogeneity the data lack varietal representativeness with respect to word structure. The languages in the sample are overwhelmingly of a familiar morphological type, organized around stems and affixes. The African languages in the sample are both Bantu languages (Swahili and Zulu), which are broadly similar to Eurasian languages with respect to displaying a concatenatively affixal strategy for morphotactic organization. The four Semitic languages in the sample show one kind of templatic morphology, but no languages in the sample use tones, reduplication, vowel length patterns, or many other types of morphological expression.

By its nature, Wiktionary only includes languages with a written form and those mostly using their practical orthography, in contrast to phonologized lexicons such as Flexique (Bonami et al., 2013). This raises several potential problems. Of particular note, orthographic systems vary widely in phonological transparency, and many orthographies neglect important distinguishing morphophonological details such as tone, segment length, and stress placement (e.g., see Parker and Sims in press): this creates problems with respect to identifying the correct inventory of forms that need to be compared. For example, the Estonian orthography underrepresents "gradation" in all but the stop consonants and, thereby, misrepresents the actual variety of contrasting forms in Estonian paradigms. Roughly speaking, Estonian consonants and vowels display a three-way contrast (short, long, and overlong) which is not represented in the orthography. This leads to the following differences in the orthographic representations versus the phonological reality for the noun *keel* 'language' (Mürk, 1997, 107):

	Orth.	**Phon.**
NOM SG	keel	ke::l
GEN SG	keele	ke:le
PART SG	keelt	ke:lt::
ILL SHORT SG	keelde	ke::le ~ ke::lde
GEN PL	keelte	ke:lte

Finally, different scripts may pose different modeling challenges, making it difficult to directly compare a model-based metric across languages written using various alphabets, abjads, syllabaries, etc.

A sample of 106 *closely related* or overlapping written languages provides a lot less information about the space of possible languages than a sample of 106 *unrelated* languages would. This is not a flaw in UniMorph per se and it does not reduce its value as a test-bed for developing morphological processors, particularly for the constrained class of variation it models. Given the limited range of morphological variation represented in UniMorph, any results concerning morphological organization beyond that sample can only support modest claims to greater generality, which themselves need to articulated into testable hypotheses. This is, of course, the same standard appropriately posed for linguistic theories that seek to motivate wide ranges of morphological organization exhibiting extraordinarily divergent strategies of surface

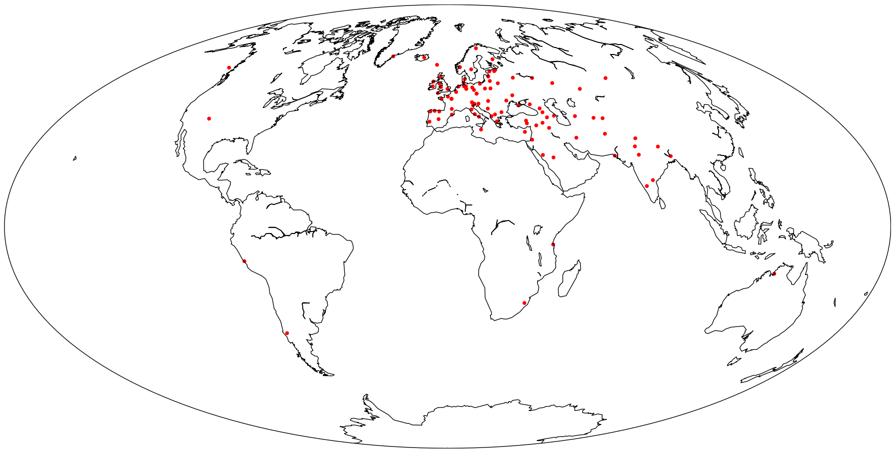


Figure 1: Geographic distribution of languages in UniMorph (languages locations from (Dryer and Haspelmath, 2013))

encoding: their credibility too is dependent on the empirical scope and reliability of the data they analyze.

3 A case study

As a concrete example, we will consider the relationship between paradigm size and predictability in morphological paradigms. Ackerman and Malouf (2013) distinguish two dimensions of morphological complexity: E-complexity (the number of affixes, allomorphs, inflection classes, etc.) and I-complexity (the interpredictability of forms in a paradigm). Ackerman and Malouf (2013) conjecture that I-complexity is what is relevant for language learnability, and that across languages E-complexity can vary widely so long as I-complexity is low enough. More recent work (Cotterell et al., 2019; Semenuks, 2019) suggests that E-complexity and I-complexity may be interrelated, and that the threshold for 'low enough' I-complexity may decrease as E-complexity increases. In what follows, we will consider some of the methodological choices that need to be made in order to properly test this claim.

For the sake of discussion, we will measure E-complexity as paradigm size, or the number of distinct feature values encoded in the database. For example, if a nominal paradigm encodes 7 cases and 2 numbers, the size of the paradigm is 14. If the paradigm size varies between lexemes, we use the most common value (i.e., the mode). To estimate I-complexity or predictability, we train a model to map a citation form and feature set to a surface form (SIGMORPHON 2016 task 1; Cotterell et al. 2016). Specifically, we use a neural encoder-decoder architecture (Kann and Schütze, 2016; Silfverberg and Hulden, 2018) implemented using OpenNMT-tf (Klein et al., 2017). Using the model, we then calculate the average per-form negative log likelihood ($-L$) of held out data.[2] The closer this value is to zero, the better the model is able to predict the correct forms. Note that we are not claiming that this is the correct way to estimate either E- or I-complexity: we have chosen it mostly because it is easy to calculate in a reproducible way. Our goal is to focus on methodological issues, not the viability of any specific linguistic analysis.

3.1 Lexicon size

One issue that immediately arises is that the performance of neural models can be highly dependent on the quantity of training data. Since there are large differences in lexicon sizes across languages in UniMorph, difference in model prediction (reflected in $-L$) may be due to training issues and not to structural differences between languages. This, of course, is important to know, since otherwise our results might be comparing incomparable phenomena.

[2] See `https://github.com/rmalouf/SCiL2020` for implementation details.

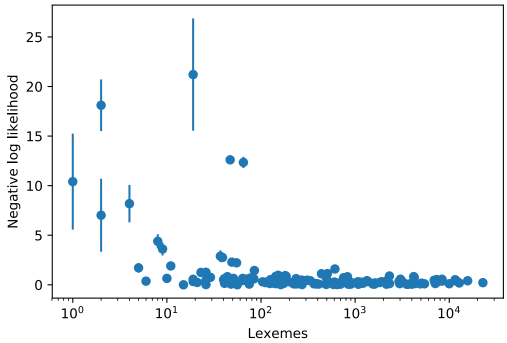


Figure 2: Negative log likelihood ($-L$) vs. lexicon size

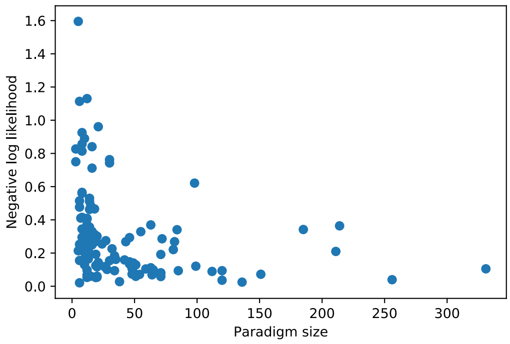


Figure 3: Negative log likelihood ($-L$) vs. paradigm size, for paradigms with >100 lexemes

To test this, we performed five-fold cross-validation to estimate $-L$ and its standard error (the standard deviation divided by $\sqrt{5}$). The results for the 87 verb paradigms and 73 noun paradigms (we exclude Tajik nouns, which list only one inflected form per lexeme) are given in Figure 2. For languages with small lexicons ($\leq$100 lexemes), we see both poor model performance (i.e., high $-L$) and high variability across train/test splits. For languages with more than 100 lexemes, however, performance looks much more consistent.

If we exclude paradigms with fewer than 100 lexemes, we are left with 55 noun paradigms and 61 verb paradigms over a total of 77 languages. The results are shown in Figure 3. At first glance, this appears to support the claim that languages can have higher I-complexity if they have low E-complexity. But, this is only true if high $-L$ is due to structural properties of the language being tested. In the following sections, we will look at a number of factors that can increase $-L$ for particular languages without any increase in I-complexity.

3.2 Overabundance

One issue that arises in examining the UniMorph data is that many (sub)paradigms permit more than a single form in a cell for a given lexeme: particular combinations of feature values can be realized by more than one exponent. For example, the past tense of English *dive* can be either *dived* or *dove*. There are several causes for this. Some examples are simply data processing errors: two distinct forms have been erroneously assigned the same feature values in extracting the data from Wiktionary. In other cases, the forms do share the same features but are not interchangeable for other reasons.

For example, the Spanish lexicon lists both *sentir* and *sentirse* as infinitive forms of the verb *sentir* 'to feel', even though the second of the two forms is (arguably) the infinitive of a different lexeme. Similarly, the Zulu verb lexicon lists both *ngiyadla* and *ngidla* as the 1st person singular present tense positive absolute form of the verb *ukudla* 'to eat'. But, these forms are not completely synonymous. The exact nature of the difference between these forms is unclear (see, e.g., Buell 2006), but they should be distinguished somehow.

The majority of cases, however, are due to genuine **overabundance**: multiple forms are listed because multiple forms are possible (Thornton, 2011, 2019). Wiktionary lists *troféen* or *trofeen* or *troféet* or *trofeet* as alternate definite singular forms of *trofé* 'trophy' in Norwegian Nynorsk, with no difference in meaning. This creates a problem for any metric which assumes that every paradigm cell has exactly one realization. This includes models evaluated using accuracy or, in our case, negative log likelihood. Using our metric, paradigms exhibiting overabundance will show higher negative log likelihood than ones that do not, for reasons that have no connection to how predictable or systematic the morphological system is.

Overall, although many languages left in the sample don't have any lexemes with multiple forms filling in a paradigm cell, it is also not rare: 18 out of 55 languages with noun paradigms and 19 out of 61 languages with verb paradigms exhibit this pattern, out of which 16 (for nouns) and 14 (for verbs) have more than multiple forms in cell for more than 10% of the lexemes. Regardless of whether the reason for this pattern is genuine overabundance or data processing errors, it nevertheless introduces difficulties into further analyses.

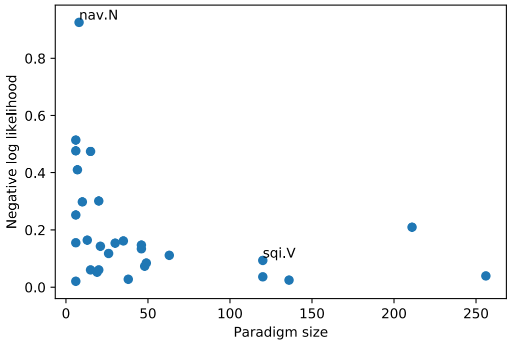


Figure 4: Negative log likelihood vs. paradigm size, for reduced sample

3.3 Defectiveness

Paradigms in the UniMorph database display many missing forms. In many cases this is due simply to incompleteness: the forms exist, but for whatever reason are not included in Wiktionary or were not extracted. However, missing forms can also be due to paradigm **defectiveness**. This is the converse of overabundance: these are paradigm cells for which there is no valid realization.

Like overabundance, missing data raises problems for any metric which assumes that every paradigm cell has exactly one realization. Forms which are missing due to incompleteness may have the effect of hurting model performance (and raising $-L$) without an underlying difference in predictability. If forms are missing due to true paradigm defectiveness, then the fact that the form is missing is something that the model needs to learn. As argued by Sims (2015), the absence of a form is as much a part of the morphological system as its presence.

3.4 Complications

To avoid modeling problems raised by overabundance and defectiveness, we can remove from the sample any paradigms with any overabundant forms and more than ten defective paradigms. This leaves 17 verb paradigms and 12 noun paradigms from 28 languages. The results for this reduced sample are shown in Figure 4 and Table 1.

The outlier in the upper left (nav.N) is Navajo nouns. The high $-L$ value for Navajo nouns is surprising, as Navajo nominal morphology is fairly straightforward. Examination of the data shows a number of inaccuracies or infelicities in the data that lead to poor model performance.

Some of the errors were introduced in the process of extracting forms from Wiktionary. The paradigm for *éé’* ‘clothes’ is shifted up one row: the 1p singular possessed form is listed as *singular* rather the correct *she’éé’*, the 2p singular is listed as *she’éé’* rather than *ne’éé’*, and so on.

Most of the problems with the Navajo nominal data, however, are consequences of the decisions made by the designers of the Navajo wiktionary. First, a brief summary of Navajo nominal morphology: nouns in Navajo form a fairly small, closed class. Inalienably possessed nouns (mostly kin relations and body parts) appear in an indefinitely possessed form (*átáá’* ‘someone’s forehead’) or with a possessive prefix (*shítáá’* ‘my forehead’). Alienably possessed nouns may appear as a bare stem (*sǫ’* ‘star’), as possessed form (*azǫ’* ‘someone’s star’), or as a possessed form with a possessive prefix (*shizǫ’* ‘my star’). The possessive prefixes show relatively little allomorphy, but the possessed form and the bare stem sometimes differ in arbitrary ways. Most Navajo nouns are unmarked for number, but a few personal nouns take a plural suffix *-ké* or *-yóó*.

The Navajo noun paradigms in Wiktionary list only the possessed forms. For alienably-possessed nouns, the bare stem (e.g., *sǫ’*) is the citation form for the lexeme but is not included in the paradigm. For inalienably-possessed nouns, the indefinite possessed form is the citation form. This inconsistency makes the two noun classes look more different than they actually are. More problematic is the fact many nouns have separate dictionary entries for possessed forms: *ké* ‘foot’ is also listed under *bikee’*, *hakee’*, and *akee’*, the 3rd person, 4th person, and indefinite possessed forms. From the model’s perspective, this looks like four separate lexemes (with four different citation forms) that happen to share the same inflected forms.

Three other high $-L$ paradigms in Table 1 are Pashto nouns, Urdu nouns, and Yiddish verbs. Like all the language samples, these paradigms are written using the practical orthography of the language. In the case of Urdu and Pashto, the writing system (based on Arabic by way of Persian) is an **abjad**: consonants are included, but many vowels are left unspecified when they should be clear to the reader from context. The Yiddish alphabet is adapted from Hebrew and is a full alphabet, but the mapping between Yiddish letters and

Language	pos	features	$-L$	s.e.	macroarea	family	genus
Albanian	V	120	0.094	0.002	Eurasia	Indo-European	Albanian
Ancient Greek	N	15	0.475	0.018	Eurasia	Indo-European	Greek
Bulgarian	V	20	0.060	0.012	Eurasia	Indo-European	Slavic
Catalan	V	48	0.073	0.002	Eurasia	Indo-European	Romance
Classical Syriac	N	13	0.164	0.112	Eurasia	Afro-Asiatic	Semitic
Crimean Tatar	N	6	0.155	0.021	Eurasia	Altaic	Turkic
Danish	V	6	0.021	0.018	Eurasia	Indo-European	Germanic
Dutch	V	15	0.060	0.006	Eurasia	Indo-European	Germanic
Estonian	N	30	0.154	0.014	Eurasia	Uralic	Finnic
Friulian	V	46	0.147	0.023	Eurasia	Indo-European	Romance
Georgian	N	19	0.052	0.006	Eurasia	Kartvelian	Kartvelian
Hebrew	N	26	0.118	0.027	Eurasia	Afro-Asiatic	Semitic
Hindi	V	211	0.210	0.116	Eurasia	Indo-European	Indic
Irish	V	63	0.111	0.010	Eurasia	Indo-European	Celtic
Lithuanian	V	49	0.084	0.010	Eurasia	Indo-European	Baltic
Lower Sorbian	V	21	0.143	0.058	Eurasia	Indo-European	Slavic
Navajo	N	8	0.925	0.317	North America	Na-Dene	Athapaskan
Occitan	V	46	0.134	0.013	Eurasia	Indo-European	Romance
Pashto	N	6	0.477	0.125	Eurasia	Indo-European	Iranian
Persian	V	136	0.025	0.006	Eurasia	Indo-European	Iranian
Quechua	N	256	0.039	0.023	South America	Quechua	Quechua
Quechua	V	38	0.028	0.016	South America	Quechua	Quechua
Romanian	V	35	0.162	0.026	Eurasia	Indo-European	Romance
Slovenian	V	20	0.301	0.042	Eurasia	Indo-European	Slavic
Tatar	N	6	0.252	0.024	Eurasia	Altaic	Turkic
Turkish	V	120	0.036	0.006	Eurasia	Altaic	Turkic
Urdu	N	6	0.514	0.107	Eurasia	Indo-European	Indic
Yiddish	V	7	0.410	0.177	Eurasia	Indo-European	Germanic

Table 1: Results for reduced sample

Unicode characters is not one-to-one. It is possible that these orthographic differences might make estimates of $-L$ difficult to compare across languages with different writing systems.

Ancient Greek nouns also have a high $-L$, but likely not for orthographic reasons. Rather, these paradigms encode overabundance using punctuation rather than multiply filled paradigm cells. For example, the genitive singular of κοῦρος 'youth' is given as "κούρου / κουροῖο / κούροιο / κουρόο / κούροο". This is presumably meant to reflect five variant forms, but the model would count that as one long (and hard to guess) form.

Another outlier, this time in the number of features, is Albanian verbs (sqi.V). According to UniMorph (and Wiktionary), each Albanian verb has 120 distinct forms. However, this number includes periphrastic tenses formed by combining an inflected verb with a particle and/or an auxiliary verb. This is a bit like counting *will have been being seen* as a distinct form of the verb 'see' in English.

The design choices embodied in Wiktionary are not necessarily incorrect. It is helpful for Navajo learners to have separate dictionary entries for prefixed forms. And, a strong argument can be made that periphrastic forms should be included as part of the paradigm in both Albanian and in English (e.g., Ackerman and Webelhuth 1998; Ackerman and Stump 2004; Bonami 2015). But, if one's goal is to use UniMorph data for cross-linguistic comparison, then these kinds of choices need to be made in a standardized way and clearly articulated. The issue is not whether data choices are right or wrong, but whether those choices are transparent and appropriate for a particular use.

3.5 Galton's problem

Even excluding Navajo nouns and the other outliers, the pattern of languages shown in Figure 4 suggests that languages in the sample with large paradigms show low $-L$. Without Navajo nouns, there are 17 verb paradigms and 11 noun paradigms from 27 languages in the sample. Is this enough to draw any conclusions about language in general?

So far, in our discussion we have used quantitative but not statistical methods. The difficulty with applying standard hypothesis testing methods to the problem is that languages that are genetically and/or areally related cannot be treated as independent observations. Of the 23 languages in the remaining sample, 16 are Indo-European and 21 are Eurasian. If the data is not analyzed using methods taking these phylogenetic and geographic proximities between the data points into account, the analyses could produce spurious correlations (Roberts and Winters, 2013). This is what Naroll (1965) calls **Galton's Problem**: the problem of making inferences based on auto-correlated observations.

Early work in quantitative typology addressed this problem through careful sample construction (Bybee, 1985; Dryer, 1988; Perkins, 1989). More recent efforts have applied hierarchical modeling techniques to control for genetic and areal affects. A survey of these techniques is beyond the scope of this paper, but see Bakker (2011) and (Bickel, 2015) for some proposals.

3.6 Construct validity

Based on the results so far, there is suggestive evidence for a relationship between the number of cells in a paradigm and $-L$ as predicted by an encoder-decoder model. The final step in any typological study has to be to show that these metrics applied in this way to this dataset connect to a relevant linguistic notion. In this case, a crucial question is whether $-L$, a measure of how well a model predicts forms, is a reasonable measure of the I-complexity of a paradigm, or how predictable forms are. This is the question of **construct validity**: does the test measure what it claims to measure?

As we said above, our goal in this paper is to highlight some of the methodological issues that come in using text databases (such as UniMorph) for typology. Our use of $-L$ is only for the sake of demonstration and we make no particular claims about its linguistic relevance. But, if this were a paper making a typological claim, then it would be essential to justify our confidence in the particular metric being used. Readers need to keep this requirement in mind when assessing and interpreting the linguistic value of results based on computational analyses of natural language data.

4 Conclusions

Large text databases open up exciting prospects for typological research, but they also create new challenges for cross-disciplinary collaboration: linguistic morphologists and typologists are practiced curators of the types of data that are most profitably investigated by new computational techniques. The previous section presented a hypothetical typological investigation using UniMorph in order to highlight some of the difficulties in carrying out such an investigation. Any work applying computational models to primary linguistic data (e.g., information-theoretic investigations of UniMorph along the lines of Cotterell et al. 2019; Pimentel et al. 2019; Wu et al. 2019) need to be carried out and evaluated with these challenges in mind. As an emergent interdisciplinary community, we should develop a set of best practices for using the resources we have and in developing a collaboratively determined direction for improving those resources.

As a start, we propose some basic requirements:

- Use UniMorph (Kirov et al., 2016, 2018) as a resource for building databases, not as a database itself: text databases should be seen as a guide for formulating directions of inquiry and identifying the types and nature of data required for systematic inquiry. The data established for this purpose must be reliable and representative for the task at hand.

- Document all choices: In order to achieve maximum transparency and replicability, all choices concerning data selection, preprocessing, representation, parsing, and modeling should be clearly specified, along with their rationales.

- Intended claims and hypotheses associated with analysis and results should be clearly articulated in order to identify their importance in the context of similar research within relevant linguistic approaches to morphological analysis. This is crucial in order to evaluate the research results from both a linguistic and computational perspective: if such results are novel, in what ways do they contribute to our understanding of natural language morphology and to the computational analysis of morphological phenomena.

- Given the cross-disciplinary nature of the relevant contributions, the vetting process for the evaluation of submissions should be distributed among linguists and computational modelers, in order to ensure research that reflects the most accurate and critical assessments from contributing fields.

References

Farrell Ackerman and Robert Malouf. 2013. Morphological organization: The low conditional entropy conjecture. *Language*, 89:429–464.

Farrell Ackerman and Gregory Stump. 2004. Paradigms and periphrasis: A study in realization-based lexicalism. In Louisa Sadler and Andrew Spencer, editors, *Projecting Morphology*, pages 111–157. CSLI Publications, Stanford.

Farrell Ackerman and Gert Webelhuth. 1998. *A Theory of Predicates*. CSLI Publications, Stanford.

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Matthew Baerman, Dunstan Brown, and Greville G. Corbett. 2002. The Surrey syncretisms database. `http://www.smg.surrey.ac.uk/syncretism/index.aspx`.

Matthew Baerman, Dunstan Brown, and Greville G. Corbett. 2005. *The Syntax-Morphology Interface: A Study of Syncretism*. Cambridge University Press.

Dik Bakker. 2011. Language sampling. In Jae Jung Song, editor, *The Oxford Handbook of Typology*. Oxford University Press.

Christian Bentz and Bodo Winter. 2013. Languages with more second language learners tend to lose nominal case. *Language Dynamics and Change*, pages 1–27.

Balthasar Bickel. 2015. Distributional typology: statistical inquiries into the dynamics of linguistic diversity. In Balthasar Bickel, Bernd Heine, and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis (2 ed.)*. Oxford University Press.

Balthasar Bickel and Johanna Nichols. 2002. Autotypologizing databases and their use in fieldwork. In *Proceedings of the International LREC Workshop on Resources and Tools in Field Linguistics, Las Palmas*.

Olivier Bonami. 2015. Periphrasis as collocation. *Morphology*, 25:63–110.

Olivier Bonami, Gauthier Caron, and Clément Plancq. 2013. Flexique: an inflectional lexicon for spoken French.

Leston Buell. 2006. The Zulu conjoint/disjoint verb alternation: focus or constituency? In Laura J. Downing, Lutz Marten, and Sabine Zerbian, editors, *Papers in Bantu grammar and description*, pages 9–30. Zentrum für Allgemeine Sprachwissenschaft, Sprachtypologie und Universalienforschung, Berlin.

Joan L. Bybee. 1985. *Morphology: A study of the relation between meaning and form*. Benjamins, Philadelphia.

Ryan Cotterell, Christo Kirov, Mans Hulden, and Jason Eisner. 2019. On the complexity and typology of inflectional morphological systems. *Transactions of the Association for Computational Linguistics*, 7:327–342.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 Shared Task—Morphological Reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin. Association for Computational Linguistics.

Michael Cysouw. 2003. *The Paradigmatic Structure of Person Marking*. Oxford University Press.

Johannes Dellert and Gerhard Jäger, editors. 2017. *NorthEuraLex (version 0.9)*.

Matthew S. Dryer. 1988. Object-verb order and adjective-noun order: Dispelling a myth. *Lingua*, pages 185–217.

Matthew S. Dryer. 1989. Large linguistic areas and language sampling. *Studies in Language*, 13(2):257–292.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Joseph H. Greenberg. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg, editor, *Universals of Grammar*, pages 73–113. MIT Press, Cambridge.

Oliver A. Iggesen. 2005. *Case-asymmetry: A World-Wide Typological Study on Lexeme-class-dependent Deviations in Morphological Case Inventories*. Lincom Europa, Muenchen.

Oliver A. Iggesen. 2013. Number of cases. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Mary Ritchie Key and Bernard Comrie, editors. 2015. *The Intercontinental Dictionary Series*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Christo Kirov, John Sylak-Glassman, Roger Que, and David Yarowsky. 2016. Very-large scale parsing and normalization of Wiktionary morphological paradigms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3121–3126.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Ryan Cotterell Ekaterina Vylomova Miikka Silfverberg Kyle Gorman, Arya D. McCarthy and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *CoNLL 2019*.

Natalia Levshina. 2016. Why we need a token-based typlogy: A case study of analytic and lexical causatives in fifteen European languages. *Folia Linguistica*, 50(2):507–542.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying universal dependencies and universal morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Harri William Mürk. 1997. *A Handbook of Estonian: Nouns, Adjectives and Verbs*. Indiana University Uralic and Altaic Series, v. 163. Indiana University, Bloomington.

Raoul Naroll. 1965. Galton's problem: The logic of cross-cultural research. *Social Research*, 32:428–451.

Jeff Parker and Andrea Sims. in press. Irregularity, paradigmatic layers, and the complexity of inflection class systems: A study of Russian nouns. In P. Arkadiev and F. Gardani, editors, *The Complexities of Morphology*. Oxford University Press.

Revere D. Perkins. 1989. Statistical techniques for determining language sample size. *Studies in Language*, 13:293–315.

Tiago Pimentel, Arya D. McCarthy, Damian Blasi, Brian Roark, and Ryan Cotterell. 2019. Meaning to form: Measuring systematicity as information. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1751–1764, Florence, Italy. Association for Computational Linguistics.

Seán Roberts and James Winters. 2013. Linguistic diversity and traffic accidents: Lessons from statistical studies of cultural traits. *PLOS ONE*, 8(8):1–13.

Arturs Semenuks. 2019. Investigating relationship between i-complexity and population size. Poster presented at the Workshop on Interaction and the Evolution of Linguistic Complexity, Edinburgh.

Steven Shearing, Christo Kirov, Huda Khayrallah, and David Yarowsky. 2018. Improving low resource machine translation using morphological glosses (non-archival extended abstract). In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 132–139, Boston, MA. Association for Machine Translation in the Americas.

Miikka Silfverberg and Mans Hulden. 2018. An encoder-decoder approach to the paradigm cell filling problem. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2889, Brussels, Belgium. Association for Computational Linguistics.

Andrea D. Sims. 2015. *Inflectional Defectiveness*. Cambridge University Press, Cambridge.

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (UniMorph schema). working draft, v. 2.

Anna M. Thornton. 2011. Overabundance (multiple forms realizing the same cell): A non-canonical phenomenon in Italian verb morphology. In Martin Maiden, John Charles Smith, Maria Goldbach, and Marc-Olivier Hinzelin, editors, *Morphological Autonomy: Perspectives From Romance Inflectional Morphology*. Oxford University Press.

Anna M. Thornton. 2019. Overabundance in morphology. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.

Bernhard Wälchli and Michael Cysouw. 2012. Lexical typology through similarity semantics: Toward a semantic map of motion verbs. *Linguistics*, 50(3):671–710.

Shijie Wu, Ryan Cotterell, and Timothy O'Donnell. 2019. Morphological irregularity correlates with frequency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5117–5126, Florence, Italy. Association for Computational Linguistics.

What Code-Switching Strategies are Effective in Dialogue Systems?

Emily Ahn[1*] **Cecilia Jimenez**[2] **Yulia Tsvetkov**[3] **Alan W Black**[3]
[1]University of Washington [2]University of Pittsburgh [3]Carnegie Mellon University
eahn@uw.edu, CCJ16@pitt.edu, {ytsvetko,awb}@cs.cmu.edu

Abstract

Since most people in the world today are multilingual (Grosjean and Li, 2013), code-switching is ubiquitous in spoken and written interactions. Paving the way for future adaptive, multilingual conversational agents, we incorporate linguistically-motivated strategies of code-switching into a rule-based goal-oriented dialogue system. We collect and release COMMONAMIGOS, a corpus of 587 human–computer text conversations between our dialogue system and human users in mixed Spanish and English. From this new corpus, we analyze the amount of elicited code-switching, preferred patterns of user code-switching, and the impact of user demographics on code-switching. Based on these exploratory findings, we give recommendations for future effective code-switching dialogue systems, highlighting user's language proficiency and gender as critical considerations.[1]

1 Introduction

Humans seamlessly adjust their communication to their interlocutors (Gallois and Giles, 2015; Bell, 1984). We adapt our language, communication style, tone and gestures; when we share more than one language with our interlocutor, we inevitably resort to multilingual production or *code-switching*—shifting from one language to another within an utterance (Sankoff and Poplack, 1981).

We envision naturalistic conversational agents that communicate fluently and multilingually as humans do. However, existing dialogue systems are agnostic to the user, generating monolingual sentences which overfit to the language, domain, and style of their training data. To enable user-centric multilingual conversational agents, dialogue systems need to be extended to accommodate and converse with bilinguals, potentially using multiple languages in an utterance, as shown in Figure 1.

Figure 1: We build a bilingual goal-oriented agent that can converse in Spanish–English code-switching with human users. In controlled settings, we collect human–computer conversations that enable us to develop effective CS strategies for future dialogue systems.

Before the rise of social media, code-switching (henceforth, CS) was primarily a spoken phenomenon, and it has been studied in spoken conversations (Lyu et al., 2010; Li and Fung, 2014; Deuchar et al., 2014). However, the spoken language domain is not directly comparable to the written one, and its spontaneous settings make it difficult to conduct controlled experiments to study accommodation in CS of one speaker to another. In controlled settings, CS has been extensively studied in psycholinguistics (Kootstra, 2012), but these are typically carefully designed experiments with few participants, which are hard to apply in large-scale data-driven scenarios like ours. In the written domain, which is the focus

*This work was done while the first author was a student at Carnegie Mellon University.

[1]This study was approved by the IRB. All code and collected data are available at https://github.com/emilyahn/commonamigos.

of our work, CS has been studied in broadcast texts such as social media (e.g. Reddit and Twitter) posts (Rabinovich et al., 2019; Aguilar et al., 2018) at the level of a single sentence and not contextualized in a dialogue.

Strikingly, little is known about human choices in written code-switching in conversations beyond the context of an individual utterance. In this paper, we introduce a novel framework which will allow us to fill this gap and study CS patterns contextualized in written conversations. Our focus languages are Spanish and English; these are often code-switched by people in Hispanic communities, who make up roughly 18% of the total US population (US Census Bureau, 2017).

We first introduce our bilingual goal-oriented dialogue system—an extension of a monolingual approach of He et al. (2017)—which controllably incorporates CS (§2). Then, we define our focus CS strategies, grounded theoretically and empirically (§3). In §4, we describe the experimental methodology and deployment of the dialogue system on crowdsourcing platforms. After collecting multilingual dialogues, we analyze patterns of CS along several axes such as the amount of CS, user accommodation (or *entrainment*) to dialogue systems that use different patterns of CS, and preferred CS patterns across user demographics (§5). Following the analysis, we provide additional background (§6) before concluding with areas for future work (§7).

Our three main contributions are (1) formulating a new task and framework of incorporating code-switching into a bilingual collaborative dialogue system. This framework has enabled us to apply and validate prior linguistic theories about CS. We show that it is useful to analyze CS along different strategies, as was suggested by Bullock et al. (2018), and we implement novel metrics to compute and generate these strategies. Our next contribution (2) is a publicly available corpus, COMMONAMIGOS, of 587 code-switched Spanish–English human–computer text dialogues and surveys, useful for further development of multilingual dialogue systems and for explorations of sociolinguistic factors of accommodation in CS (cf. Danescu-Niculescu-Mizil et al., 2011). Finally, (3) our exploratory analyses of CS patterns in this corpus serve as a crucial first step to enable naturalistic bilingual dialogue systems in the future.

2 Bilingual Collaborative Human–Computer Dialogue System

Our ultimate goal is to study human preferences in written code-switching, and to integrate this knowledge into bilingual, adaptive dialogue systems. To gain insights into human CS patterns and to enable such systems, however, we first need to collect examples of multilingual human–computer dialogues, a resource that does not yet exist.

To collect human–computer dialogues in a controlled manner, we (1) modify an existing goal-oriented dialogue framework to code-switch; (2) create multiple instances of code-switching dialogue systems, where each instance follows one pre-defined strategy of CS as described in §3; and (3) analyze collected dialogues and study how people communicate differently with dialogue agents following a particular strategy.

We begin by modifying an existing goal-oriented collaborative dialogue framework (He et al., 2017). The framework implements a scenario of discussing mutual friends given a knowledge base, private to each interlocutor. Each of the interlocutors has a list of friends with attributes such as hobby and major. Only one friend is the same across both lists, and the goal is to find that mutual friend via collaborative discussion over text chat.

We extend this framework to a bilingual Spanish–English goal-oriented collaborative dialogue. In our bilingual interface, users see the private table of friends and attributes in both Spanish and English.

To code-switch in language generation, we add modifications (visualized in green in Figure 2) to the original monolingual generation (in blue). The rule-based agent generates English strings, which are passed to an Automatic Machine Translation (MT) system[2] in order to receive the Spanish translations. With parallel English and Spanish utterances, we define rules and templates to output a bilingual utterance following one of the CS strategies described in §3 for the full duration of the chat (see examples in Table 1).

To process text from the users, utterances are first passed to the MT whose target language is English. The monolingual dialogue system receives English strings and parses utterances into basic entities, and this informs the next turn from the dia-

[2]We use Google Translate API, a state-of-the-art MT that produced reliable translations.

Strategy		Example Sentence	Miami	Twitter
Monolingual	*EN*	Do you have any friend who studies linguistics?	–	–
	SP	*¿Tienes algún amigo que estudie lingüística?*	–	–
Insertional	$SP \xrightarrow{ins} EN$	Do you have any *amigo* who studies *lingüística*?	9.0%	5.5%
	$EN \xrightarrow{ins} SP$	*¿Tienes algún* friend *que estudie* linguistics?	25.7%	30.1%
Alternational	$EN \xrightarrow{alt} SP$	Do you have any friend *que estudie lingüística*?	12.2%	12.0%
	$SP \xrightarrow{alt} EN$	*Tienes algún amigo* that studies linguistics?	15.7%	10.5%
Informal	$+ EN \xrightarrow{ins} SP$	hey *tienes algún* friend *que estudie* linguistics?	–	–
	$+ SP \xrightarrow{alt} EN$	*pues tienes algún amigo* that studies linguistics?	–	–
Neither	–	*pero* she is the case manager for those patients	37.5%	41.9%

Table 1: We show transformations of the same example sentence (references first given monolingually) in each CS strategy, as would be generated by our dialogue system. The example for Neither is from the Miami corpus and is not an utterance we generate. Note that the Informal setting can be added to either Insertional or Alternational strategies, so 2 of the possible 4 informal settings are given in this set. We also verify that our two main strategies have a presence in existing corpora (Miami and Twitter).

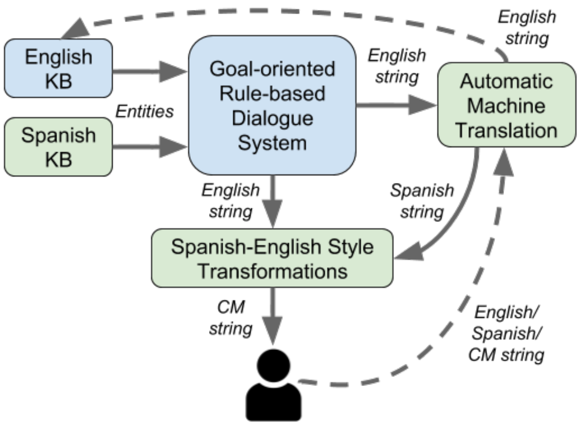


Figure 2: We add bilingual adaptations (in green) to the existing monolingual rule-based generation (in blue). The main dialogue system generates code-switched text via MT and a set of linguistically-informed code-switching rules. It receives the user's (code-switched) text after it was translated into English.

logue agent.

3 Code-Switching Strategies

We explore a variety of code-switching strategies and integrate these in our bilingual dialogue systems; each system follows one pre-defined strategy throughout the whole conversation. In this section, we describe the strategies we use, and how we operationalize them to detect and generate varied CS utterances in our dialogue system. We also verify the prevalence of these strategies in Spanish–English corpora in related domains: the Miami corpus of transcribed spontaneous speech (Deuchar et al., 2014), and a Twitter corpus (Molina et al., 2016). Examples of an utterance in each strategy along with the distribution of these strategies in both Twitter and Miami corpora are given in Table 1.

We follow Muysken's (2000) approach. The first strategy from Muysken (2000) is **Insertional code-switching**, which follows the Myers-Scotton framework of a Matrix Language (MatL) and an Embedded Language (EmbL). The structure and grammar of the MatL is maintained while inserting the EmbL (often single words or phrases) in certain spots (Myers-Scotton, 1993). According to Joshi (1982), closed class items such as determiners, quantifiers, etc., would remain in the MatL. This has also been shown to be more commonly used when the speakers are not equally proficient in both languages (Deuchar et al., 2007).

We experiment with two conditions: (1) retaining the grammar of English while inserting Spanish nouns ($SP \xrightarrow{ins} EN$), and (2) using Spanish grammar while inserting English nouns ($EN \xrightarrow{ins} SP$).

Next, we experiment with **Alternational code-switching**, when the two languages remain more separate and alternate after clauses. Switch-points adhere to constituent boundaries (Sankoff and Poplack, 1981) and can separate topics or sentences (Ardila, 2005). This has been shown to be more prevalent among fluent or highly proficient bilinguals as a form of more stable bilingualism (Deuchar et al., 2007).

We again experiment with two conditions, either beginning in English for a phrase and then switching to Spanish ($EN \xrightarrow{alt} SP$), or beginning in Spanish and then switching to English ($SP \xrightarrow{alt} EN$).

Since people may code-switch more often in informal, casual settings or when there is higher rapport, we experiment with the above four CS strategies with our agent speaking either informally or formally. We modulate formality by adding discourse markers. Discourse markers are known to be actively used by speakers in improving the flow of dialogue, and they remain relatively independent of syntax or semantics (Schiffrin, 1988). Within CS speech, these markers can be adopted as an easy form of lexical borrowing by bilinguals of varying proficiency. In particular, Spanish markers within English speech can be used to signify a less formal tone or to reveal Latino social identity (Torres, 2011). Therefore we define our agent's informal setting (+*Informal*) to have discourse markers added to either Insertional CS or Alternational CS utterances.

3.1 Detecting Insertional and Alternational Code-Switching

The two strategies can be manually detected by linguists, but there has not been a direct attempt to automatically label CS utterances as Insertional or Alternational.[3] We therefore introduce a novel method to computationally classify CS utterances into $EN \xrightarrow{alt} SP$, $SP \xrightarrow{alt} EN$, $SP \xrightarrow{ins} EN$, $EN \xrightarrow{ins} SP$, or *Neither*.[4]

An utterance is Alternational when it switches from Lang_A to Lang_B under 2 conditions: (1) there is a contiguous span of 2+ words in Lang_A followed by a contiguous span of 2+ words in Lang_B, and (2) there is at least 1 finite (i.e. conjugated) verb form or auxiliary word in each language.[5]

If the utterance is not first classified as Alternational, it is next tested for Insertional. We define Insertional CS to occur under 3 conditions: (1) the MatL has at least 1 function word or finite verb, (2) the EmbL has at least one content word (either a noun or an adjective), and (3) the MatL has more tokens than the EmbL. This metric ensures maintaining the grammar of the MatL with insertions of the EmbL.

We test our implementation of this metric on a gold set of 150 CS utterances (50 each from Miami, Twitter, and CommonAmigos datasets) annotated for strategy jointly by two linguists proficient in both Spanish and English. A third linguist achieves a Cohen's κ of 0.75 (substantial agreement) or an F1 of 0.8 against the adjudicated gold set. Our implementation receives an F1 of 0.76 on the same gold set.

To verify the coverage of these types of CS, we analyze their prevalence in the Miami and Twitter corpora, with distributions given in Table 1. We observe that the most commonly used strategy is Insertional CS, specifically $EN \xrightarrow{ins} SP$, which mirrors findings from a Spanish–English corpus of blogs from Montes-Alcalá (2007).

4 Data Collection

In order to examine effects of different CS strategies with human bilingual speakers, we modify an existing dialogue system (§2) and deploy it to chat with online crowdworkers.

4.1 Crowdsourcing

We release this task on two crowdsourcing platforms: Amazon Mechanical Turk and Figure Eight.[6] In order to target Spanish–English bilinguals, we limit workers to be in the US,[7] and then include several ungraded Spanish proficiency test questions.[8]

Additionally, the introduction and instructions to the task are purely written in Spanish to prime the user in both languages, given that English is usually the default language for tasks released in the US. For each chat, there are always 10 friends with 3 attributes each (randomly selected with varying complexity). Users have up to 8 minutes to complete the task. Besides the 8 CS conditions, we have 2 more monolingual conditions (Spanish and English), as well as a Random CS condition where a switch point could occur with 50% chance at every smallest word unit.

[3] Bullock et al. (2018) gathered metrics to identify those two strategies across an entire corpus but not across a single utterance.

[4] This method has been refined after several iterations of discussions with linguists and examining the implementation's coverage over annotations.

[5] Detecting verbs and auxiliaries was made possible by generating English and Spanish POS tags from Spacy, available at https://spacy.io/.

[6] https://www.mturk.com; https://www.figure-eight.com.

[7] Other countries were not included in order to limit the variance of cultural factors for Spanish–English CS.

[8] 92% of all users scored 67%+ accuracy on 3 questions.

# Dialogues	587
% Extrinsic task success	64%
Avg # user utterances	7.9
Avg # tokens / utterance	6.2
EN vocab size	571
SP vocab size	846
% EN utterances	16%
% SP utterances	44%
% CS utterances	39%
% dialogues w/ CS	70%

Table 2: COMMONAMIGOS, our bilingual corpus of crowdsourced chats, has a strong presence of CS.

4.2 Collected Dialogues

We report general statistics of our collected dialogues in Table 2.

A total of 737 dialogues are collected, but 587 remain for analysis after removing chats with missing text or surveys from users. From the pool of 587 valid chats, there are 296 unique workers because some did more than one task. The self-reported survey reveals that the mean age of the workers is 31, 60% of them are male, and the most frequently reported countries of origin are USA, Venezuela, and Mexico.

Examples of conversations gathered with crowdsourced bilinguals are given in Table 3. An interesting observation is that the user chooses to emulate the strategy instead of echoing that lexical item in the $SP \xrightarrow{alt} EN$ Alternational condition. Even when the agent uses the Spanish word *contabilidad*, the user says the equivalent meaning in English, which is *accounting*. Similarly, when the $SP \xrightarrow{alt} EN$ agent discusses *dancing*, the user replies with the Spanish equivalent, *bailar*, thus prioritizing strategy over lexicon.

5 Analysis

We examine the subtleties of how users code-switched under different conditions, and share our main findings below. The questions we now explore are how much do the users code-switch, how do they do it, and how do agent strategies factor into response style?

5.1 Our bilingual dialogue system elicits code-switching

Our first encouraging finding is that a high majority of dialogues contain CS from the user (Table 2), although the users were not explicitly required to code-switch. This implies that CS is a prevalent communication style and that conversational agents could benefit from supporting multilinguality.

We first analyze the amount or presence of CS from the users. Guzmán et al. (2017) defined several metrics based on quantifying token counts and span lengths of continuous monolingual tokens. The Multilingual-index (**M-idx**) reflects how balanced the tokens are in each language, where 0 is fully monolingual and 1 is an equal number of tokens per language. The Integration-index (**I-idx**) is the probability of switching languages between any two tokens, where 0 is fully monolingual and 1 is a perfectly interleaved corpus, with a switch at every word.[9] Higher values of both indices imply a higher quantity of CS.

Table 4 shows that $SP \xrightarrow{ins} EN$ *+Informal* and Alternational conditions result in higher M-indices than average. Most notably, the $EN \xrightarrow{ins} SP$ condition results in the lowest M-idx and I-idx from users. We reason that this is due to receiving more monolingual Spanish text from users than in any other condition, a potential result of having the crowdworkers primed to be in Spanish mode. Conversely, the $SP \xrightarrow{ins} EN$ conditions maintain markedly high CS indices from users. $SP \xrightarrow{ins} EN$, the agent with the highest number of English tokens, could have encouraged users to balance their Spanish tokens with more English. We advise future CS systems to be aware of their target audience's assumptions of the agent's default language.

The added formality setting has a number of effects on the two main strategies. Across all 4 Insertional and Alternational conditions, *+Informal* reduces the average number of tokens in a user's reply (seen in Table 4), which could be a result of users being more casual with the dialogue system. M-idx increases for both Insertional strategies while sharply decreasing for both Alternational strategies. I-idx slightly increases for all strategies except $SP \xrightarrow{ins} EN$. We can recommend that if the goal of a future CS dialogue system is to be efficient in number of turns, the $SP \xrightarrow{ins} EN$ strategy is useful, but if the goal is to keep the user engaged and chat for longer, $SP \xrightarrow{ins} EN$ *+Informal*

[9]To calculate I-idx in a given dialogue, all utterances by one party are concatenated in order, so switch-points can occur across utterance boundaries.

EN $\xrightarrow{alt}$ ***SP***	***EN*** $\xrightarrow{ins}$ ***SP***
A: I have 2 friends *que estudiaron la contabilidad* [that studied accounting] H: *yo tambien* [me too]. one that studies accounting *trabaja en el concesionario de coches y el otro en la oficina* [works at the car dealership and the other in the office] A: Do you have any friend who likes dancing *o amigos a los que les guste hornear* [or friends who like to bake]? H: *nadie le gusta bailar* [no one likes to dance]. one likes baking–*el/ella estudia fisica* [he/she studies physics]	A: *¿Tiene* [Do you have] friends *que trabajen en el* [who work at the] theater *o un* [or a] friend *que trabaje en la* [that works at the] jewelry store ? H: *si. la del* [yes. the one from] jewelry store *le gusta dormir* [likes to sleep] A: *tengo* [I have] 1 friend *que le gusta* [who likes] acting, 1 friend *que trabaja en el* [who works at the] zoo H: *la del teatro le gusta* [the one from the theater likes] photography
SP $\xrightarrow{alt}$ ***EN +Informal***	***SP*** $\xrightarrow{ins}$ ***EN +Informal***
A: *tengo un amigo* [I have a friend] who studied english.. *y tú* [and you]? H: *no tengo... solo tengo un amigo que estudio* [I don't have... I only have a friend that studied] linguistics A: hey *tengo dos amigos* [I have two friends] who like sewing H: *yo tengo un amigo que le gusta* [I have a friend that likes] sewing!	A: do you have any *amigos* [friends] who studied *derecho* [law] ? H: no i don't H: *tienes un amigo a quien le gusta cocinar* [do you have a friend who likes to cook]? A: nah i have no *amigo* [friend] who likes *cocinar* [to cook]..

Table 3: These examples from our corpus of human (H) interactions with rule-based CS agents (A) show a diversity of CS strategies, given the static agent strategy in **bold**.

or *SP* $\xrightarrow{alt}$ *EN +Informal* could yield more turns. We encourage CS dialogue systems to consider implementing casual styles of speech in CS, as our simple additions of discourse markers produced patterned changes in token length and amount of CS.

5.2 Agent strategy can affect user strategy

We see the presence of entrainment between agent strategy (condition) and user strategy. In the matrix in Figure 3, perfect entrainment (where all the users' CS utterances use the same fixed agent strategy) would be shown with a normalized value of 1.0 along the diagonal. We compare values across CS conditions (without examining +*Informal* for now) to the random baseline, which ideally reveals the natural unconditioned distribution of user strategy.[10] Because the values on the diagonal are significantly greater than in the random condition ($p < .05$), we conclude that the agent's strategy had influence on the user's code-switching.

[10]Reassuringly, the percentages in this random condition are similar to the distribution of the Miami and Twitter corpora from Table 1.

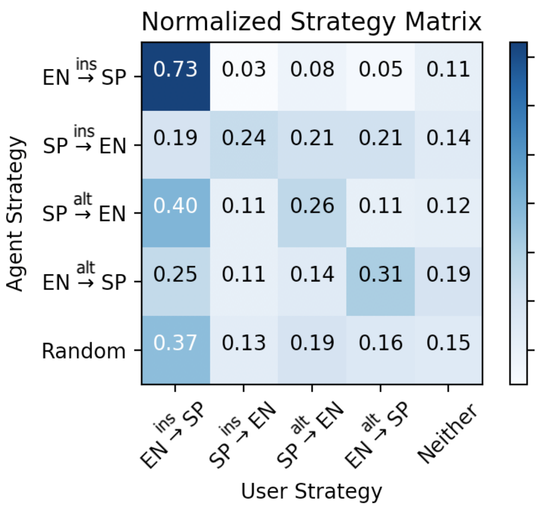


Figure 3: We find entrainment in our data. Given each agent strategy condition (per row), we display the normalized distribution of which strategies the users used (only accounting for utterances that are code-switched). Darker colors along the major diagonal indicate complete entrainment, and the random agent strategy at the bottom is shown for comparison.

For conditions where English is the main (or

Agent	# Dial	% Success	Avg Utts	Avg Tok	% CS Dial	% CS Utts	M-idx	I-idx
Average	53.4	64	7.9	6.2	70	39	0.74	0.23
Std Dev	(7.8)	(11)	(0.9)	(0.4)	(8)	(8)	(0.20)	(0.04)
$EN \xrightarrow{ins} SP$	**70**	**47**	8.4	6.3	74	42	**0.51**	0.23
+Informal	**44**	**77**	7.4	**5.7**	**80**	44	0.57	0.26
$SP \xrightarrow{ins} EN$	58	62	7.2	**6.9**	74	**52**	0.93	0.26
+ Informal	**44**	64	8.6	6.0	75	37	**0.99**	0.26
$SP \xrightarrow{alt} EN$	54	74	7.5	6.4	76	39	0.88	0.24
+Informal	56	**45**	**9.7**	6.1	75	40	0.71	0.26
$EN \xrightarrow{alt} SP$	55	**76**	7.9	6.3	71	40	0.91	0.23
+Informal	47	64	7.7	6.1	72	37	0.70	0.23
Mono SP	46	72	7.2	6.1	**57**	**26**	**0.37**	**0.16**
Mono EN	54	69	**6.4**	6.5	**54**	**25**	0.74	**0.16**
Random	59	64	8.2	**5.3**	66	39	0.86	0.22

Table 4: These general statistics show dialogue quantity, length, and extrinsic success of users, as well as user quantity of CS under different agent strategies. Values further than 1 standard deviation away from the mean are in **bold**.

starting) MatL, $EN \xrightarrow{ins} SP$ occurs less often, while other English-based CS strategies are used more often. There is also more sensitivity to the specific English strategy because more utterances are classified as $SP \xrightarrow{ins} EN$ in $SP \xrightarrow{ins} EN$ conditions and $EN \xrightarrow{alt} SP$ in $EN \xrightarrow{alt} SP$ conditions. Overall, $EN \xrightarrow{ins} SP$ is the most popular strategy used—it is most common in the $EN \xrightarrow{ins} SP$ condition, but it still keeps a strong presence in other conditions. We recommend $EN \xrightarrow{ins} SP$ to be a good default strategy in future CS agents, as that also follows the prevalent styles in the Miami and Twitter corpora (§3.1).

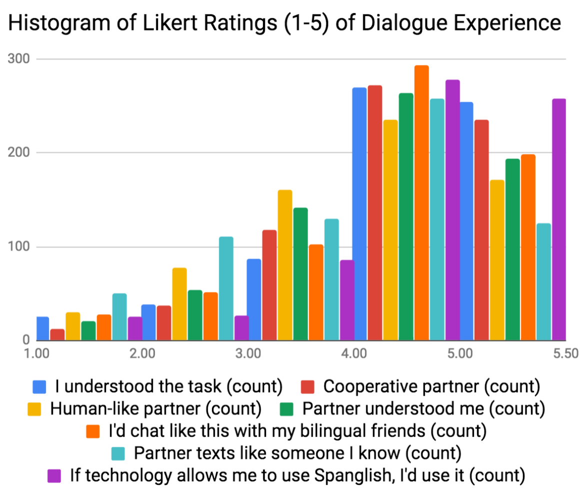


Figure 4: As an aggregate, users have generally positive experiences with our CS agent. They would rate their agreement with statements given in the legend, where 1 = Strongly Disagree, and 5 = Strongly Agree.

5.3 Users succeed in their dialogues

We define two types of success in the dialogues: (1) Extrinsic success (the binary task of finding the mutual friend in 8 minutes), and (2) User experience (self-reported measures on an agreement scale of 1-5, e.g. *"I understood the task perfectly"*, or *"My task partner texts like someone I know"*).

From Table 4, all Alternational and monolingual conditions achieve consistently high rates of extrinsic task success. This could reveal that longer spans of monolingual tokens aid in users comprehending the task, so we recommend CS systems to adhere to Alternational strategies if they desire specific goals to be achieved. As for user experience, Figure 4 displays users generally agreeing with statements such as *"I'd chat like this with my bilingual friends"*. Full exploration of variables affecting these ratings can be done with our CommonAmigos corpus. Regarding entrainment, we do not find significant correlations with any type of success metric.

5.4 User demographics affect CS

Beyond analysis of the aggregate data, we find strong effects of the following user attributes.

Language Proficiency Our findings support the hypothesis from Deuchar et al. (2007) in that more proficient bilinguals (balanced in both languages) use Alternational strategies more often than asymmetrical bilinguals. We examine this by binning the groups into three categories from the self-reported language ability metric: highly proficient in both English and Spanish, dominant English only, and dominant Spanish only.[11] Compared to the aggregate report of user CS, dominant English speakers use $SP \xrightarrow{ins} EN$ more heavily, while dominant Spanish speakers use $EN \xrightarrow{ins} SP$ more heavily. Alternational CS occurs in those two groups but is more present in the balanced bilingual group.

For the dominant English speakers, a higher M-idx correlates with better agreement on statements such as *"My task partner was very cooperative"*. When these users entrain more to the agent's CS strategy, the number of turns in the dialogue also increases. Also, even though their extrinsic task success is low in the monolingual Spanish condition, almost all CS conditions boosted task success. Together, these findings show that the dialogue experience overall improves for less-balanced bilinguals when the agent uses CS instead of their weaker monolingual language. This supports a line of pedagogy that advocates incorporation of CS in second language instruction (cf. Moore, 2002).

Gender Reported gender[12] yields strong correlations in user CS strategy. When females chat with higher M-idx and I-idx values, they agree more with the statement *"I am very likely to chat like I did in this task when messaging with my bilingual friends"*. Under informal conditions, females also have longer dialogues, a higher percentage of CS utterances, and a higher percentage of dialogues containing any CS—all of which prove to be an opposite effect for males. These findings reflect that females may code-switch more naturally and will respond better to more informal CS dialogue systems.

6 Related Work

We provide a brief overview of previous works in the domains of CS and dialogue.

Most closely related to ours is the work of Ramanarayanan and Suendermann-Oeft (2017) who introduced a chatbot that spoke from a fixed set of Spanish–English and Hindi–English machine prompts to encourage human bilinguals to code-switch back to the agent. Our work takes this interaction further and does not assume a restricted set of sentences. Rather, we control one side of the spontaneous dialogue based on different CS strategies in order to learn human preferences when code-switching.

Sitaram et al. (2019) have surveyed attempts to integrate CS into NLP and Speech processing domains. These domains include Part-of-Speech tagging (Solorio and Liu, 2008; Soto and Hirschberg, 2018), Language Identification (Ramanarayanan and Pugh, 2018; Rijhwani et al., 2017), Named Entity Recognition (Aguilar et al., 2018), Language Modeling (Chandu et al., 2018b), Automatic Speech Recognition (ASR) (Yilmaz et al., 2018), and Speech Synthesis (Rallabandi and Black, 2017). There also has been a push to generate CS datasets synthetically to improve CS language modeling (Pratapa et al., 2018), or manually crowdsource CS utterances towards CS Question–Answering and dialogue systems (Chandu et al., 2018a; Banerjee et al., 2018).

Various other research has centered around understanding when and why people code-switch. Linguistically-driven methods have found that cognates and acoustic cues allow for more fluid switching between the languages (Kootstra et al., 2012; Fricke et al., 2016).

When pertaining to a dialogue setting, CS has been found to fulfill different goals of speakers (Begum et al., 2016). Solorio and Liu (2008) discussed how sociopragmatic factors, such as the topic being discussed and the rapport between the speakers, could influence the style of CS. Additionally, choosing to use one language over another can be a pragmatic way to mark sentiment, as Rudra et al. (2016) found in Hindi–English Twitter data. These findings support our aim of understanding CS in nuanced contexts of dialogue.

In dialogue generally, entrainment between conversational partners has been shown to improve task success and perceived naturalness (Reitter and Moore, 2014; Nenkova et al., 2008). In bilingual settings, accommodation has been recorded since Giles et al. (1973), where French–English speakers would choose their language according to their audience. More recently in entrainment of CS, Soto et al. (2018) showed a con-

[11] This is the strongest among various weak signals indicating language proficiency, namely the Spanish proficiency quiz, reported age of acquisition for each language, country of origin, and frequency of language use.

[12] "Other" gender constitutes 1% of users and is set aside for this analysis.

vergence in the quantity of CS between speakers over the course of long conversations in the Miami data. Fricke and Kootstra (2016) also found that the presence of CS can affect the utterance following it. Our work is the first to identify entrainment of diverse CS strategies beyond language choice in Bawa et al. (2018).

7 Conclusion

Through our novel Spanish–English dialogue framework, we generate code-switching utterances to which bilingual users also respond in various forms of code-switching. We find that users sometimes adapt to the agent's code-switching, but their choice of CS strategy primarily depends on their bilingual language proficiency. Adding discourse markers to make the agent less formal also affects patterns of user CS among female participants. Finally, extrinsic task success is not significantly affected by CS strategy, though users indicated positive dialogue experiences.

There are numerous follow-up directions that can be taken with our framework and with the novel COMMONAMIGOS corpus. For example, analyses can be done on the types of switch points, investigating attributes such as simplicity or frequency of the word that is switched, the nature of it being a cognate (Soto et al., 2018), or even the cognitive accessibility of switch words from users' mental lexicons.

We acknowledge that COMMONAMIGOS reflects a specific population of users that would not represent all Spanish–English speakers across the world, and the crowdworker population may also be skewed in ways we cannot identify. Future work should consider other groups of Spanish–English speakers, as well as other language pairs such as Hindi–English or Tagalog–English, in order to learn how these varieties may be linguistically or functionally comparative to our findings.

The implications of our current work, which reveal which CS strategies are more entrainable than others, could help CS agents adapt to users and to better parse and predict user utterances with a more informed CS language model.[13] Future agents should incorporate different CS strategies dynamically within a single conversation that entrain to the user. In order to move beyond a rule-based agent, in future work we can leverage neural language generation systems (e.g., Park and Tsvetkov, 2019) trained on CS data. From here, we can usher in an era of bilingual dialogue systems that brings human–computer interactions to a more personalized space.

[13]This approach is similar to a method where ASR systems that lexically entrain users can lower ASR error rates (Levitan, 2013).

8 Acknowledgments

We acknowledge helpful input from the anonymous reviewers. We also thank Gayatri Bhat, Cindy Blanco, Anjalie Field, Melinda Fricke, Shirley Anugrah Hayati, He He, Sachin Kumar, Chan Young Park, Anat Prior, Sai Krishna Rallabandi, Shruti Rijhwani, Shuly Wintner, and Yiheng Zhou for fruitful discussions. Finally, we sincerely thank our annotator, Joshua Baumgarten. This work was supported by NSF grant IIS-1812327 and NSF GRFP grant DGE1745016.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Named entity recognition on code-switched data: Overview of the CALCS 2018 shared task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147.

Alfredo Ardila. 2005. Spanglish: An Anglicized Spanish Dialect. *Hispanic Journal of Behavioral Sciences*, 27(1):60–81.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. *arXiv preprint arXiv:1806.05997*.

Anshul Bawa, Monojit Choudhury, and Kalika Bali. 2018. Accommodation of conversational code-choice. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 82–91.

Rafiya Begum, Kalika Bali, Monojit Choudhury, Koustav Rudra, and Niloy Ganguly. 2016. Functions of code-switching in tweets: An annotation scheme and some initial experiments. *LREC. i*, pages 1644–1650.

Allan Bell. 1984. Language style as audience design. *Language in society*, 13(2):145–204.

Barbara E Bullock, Gualberto Guzmán, Jacqueline Serigos, and Almeida Jacqueline Toribio. 2018. Should code-switching models be asymmetric? *Proc. Interspeech 2018*, pages 2534–2538.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neuman, Manoj Chinnakotla, Eric Nyberg, and Alan W Black. 2018a. Code-mixed question answering challenge: Crowdsourcing data and techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38.

Khyathi Chandu, Thomas Manzini, Sumeet Singh, and Alan W Black. 2018b. Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 92–97.

Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. 2011. Mark my words!: linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web*, pages 745–754. ACM.

Margaret Deuchar, Peredur Davies, Jon Herring, M Carmen Parafita Couto, and Diana Carter. 2014. Building bilingual corpora. *Advances in the Study of Bilingualism*, pages 93–111.

Margaret Deuchar, Pieter Muysken, and Sung Lan Wang. 2007. Structured variation in codeswitching: Towards an empirically based typology of bilingual speech patterns. *International Journal of Bilingual Education and Bilingualism*, 10(3):298–340.

Melinda Fricke and Gerrit Jan Kootstra. 2016. Primed codeswitching in spontaneous bilingual dialogue. *Journal of Memory and Language*, 91:181–201.

Melinda Fricke, Judith F Kroll, and Paola E Dussias. 2016. Phonetic variation in bilingual speech: A lens for studying the production-comprehension link. *Journal of Memory and Language*, 89:110–137.

Cindy Gallois and Howard Giles. 2015. Communication accommodation theory. *The international encyclopedia of language and social interaction*, pages 1–18.

Howard Giles, Donald M Taylor, and Richard Bourhis. 1973. Towards a theory of interpersonal accommodation through language: Some canadian data. *Language in society*, 2(2):177–192.

François Grosjean and Ping Li. 2013. *The Psycholinguistics of Bilingualism.* Wiley-Blackwell.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *Proc. Interspeech 2017*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1766–1776.

Aravind K Joshi. 1982. Processing of sentences with intra-sentential code-switching. *Proceedings of the 9th conference on Computational Linguistics*, 1:145–150.

Gerrit Jan Kootstra. 2012. *Code-switching in monologue and dialogue: Activation and alignment in bilingual language production.* [Sl: sn].

Gerrit Jan Kootstra, Janet G Van Hell, and Ton Dijkstra. 2012. Priming of code-switches in sentences: The role of lexical repetition, cognates, and language proficiency. *Bilingualism: Language and Cognition*, 15(4):797–819.

Rivka Levitan. 2013. Entrainment in spoken dialogue systems: Adopting, predicting and influencing user behavior. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 84–90.

Ying Li and Pascale Fung. 2014. Language Modeling with Functional Head Constraint for Code Switching Speech Recognition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 907–916.

Dau-Cheng Lyu, Tien Ping Tan, Engsiong Chng, and Haizhou Li. 2010. SEAME: a mandarin-english code-switching speech corpus in south-east asia. In *INTERSPEECH 2010*, pages 1986–1989.

Giovanni Molina, Nicolas Rey-Villamizar, Thamar Solorio, Fahad Al-Ghamdi, Mahmoud Gohneim, Mona Diab, and Abdelati Hawwari. 2016. Overview for the Second Shared Task on Language Identification in Code-Switched Data. *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Cecilia Montes-Alcalá. 2007. Blogging in Two Languages : Code-Switching in Bilingual Blogs. *Selected Proceedings of the Third Workshop on Spanish Sociolinguistics*, pages 162–170.

Danièle Moore. 2002. Code-switching and learning in the classroom. *International journal of bilingual education and bilingualism*, 5(5):279–293.

Pieter Muysken. 2000. Bilingual speech: a typology of code-mixing.

Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching. *Language in society*, 22(4):475–503.

Ani Nenkova, Agustin Gravano, and Julia Hirschberg. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies: Short papers*, pages 169–172. Association for Computational Linguistics.

Chan Young Park and Yulia Tsvetkov. 2019. Learning to generate word- and phrase-embeddings for efficient phrase-based neural machine translation. In

Proceedings of the 3rd Workshop on Neural Machine Translation and Generation.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1543–1553.

Ella Rabinovich, Masih Sultani, and Suzanne Stevenson. 2019. Codeswitch-reddit: Exploration of written multilingual discourse in online discussion forums. In *Proc. of EMNLP*.

Saikrishna Rallabandi and Alan W Black. 2017. On Building Mixed Lingual Speech Synthesis Systems. In *Interspeech 2017*, pages 52–56.

Vikram Ramanarayanan and Robert Pugh. 2018. Automatic token and turn level language identification for code-switched text dialog: An analysis across language pairs and corpora. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 80–88.

Vikram Ramanarayanan and David Suendermann-Oeft. 2017. Jee haan, i'd like both, por favor: Elicitation of a code-switched corpus of hindi–english and spanish–english human–machine dialog. In *Proc. Interspeech 2017*, pages 47–51.

David Reitter and Johanna D Moore. 2014. Alignment and task success in spoken dialogue. *Journal of Memory and Language*, 76:29–46.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Sekhar Maddila. 2017. Estimating Code-Switching on Twitter with a Novel Generalized Word-Level Language Detection Technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1971–1982.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do hindi-english speakers do on twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141.

David Sankoff and Shana Poplack. 1981. A formal grammar for code-switching. *Research on Language & Social Interaction*, 14(1):3–45.

Deborah Schiffrin. 1988. *Discourse markers*. 5. Cambridge University Press.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Thamar Solorio and Yang Liu. 2008. Learning to predict code-switching points. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, (October):973–981.

Victor Soto, Nishmar Cestero, and Julia Hirschberg. 2018. The role of cognate words, POS tags, and entrainment in code-switching. *Proc. Interspeech 2018*, pages 1938–1942.

Victor Soto and Julia Hirschberg. 2018. Joint part-of-speech and language ID tagging for code-switched data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–10.

Lourdes Torres. 2011. Spanish in the United States: Bilingual Discourse Markers. In *The Handbook of Hispanic Sociolinguistics*.

US Census Bureau. 2017. B03003 - hispanic or latino origin. American Community Survey 1-Year Estimates.

E Yilmaz, A Biswas, E Van der Westhuizen, F De Wet, and T Niesler. 2018. Building a unified code-switching ASR system for south african languages. In *Proc. Interspeech 2018*.

MonaLog: a Lightweight System for Natural Language Inference Based on Monotonicity

Hai Hu† **Qi Chen**† **Kyle Richardson**‡
Atreyee Mukherjee† **Lawrence S. Moss**† **Sandra Kübler**†
†Indiana University, Bloomington, IN, USA
‡Allen Institute for Artificial Intelligence, Seattle, WA, USA
{huhai,qc5,atremukh,lmoss,skuebler}@indiana.edu
kyler@allenai.org

Abstract

We present a new logic-based inference engine for natural language inference (NLI) called MonaLog, which is based on natural logic and the monotonicity calculus. In contrast to existing logic-based approaches, our system is intentionally designed to be as lightweight as possible, and operates using a small set of well-known (surface-level) monotonicity facts about quantifiers, lexical items and token-level polarity information. Despite its simplicity, we find our approach to be competitive with other logic-based NLI models on the SICK benchmark. We also use MonaLog in combination with the current state-of-the-art model BERT in a variety of settings, including for compositional data augmentation. We show that MonaLog is capable of generating large amounts of high-quality training data for BERT, improving its accuracy on SICK.

1 Introduction

There has been rapid progress on natural language inference (NLI) in the last several years, due in large part to recent advances in neural modeling (Conneau et al., 2017) and the introduction of several new large-scale inference datasets (Marelli et al., 2014; Bowman et al., 2015; Williams et al., 2018; Khot et al., 2018). Given the high performance of current state-of-the-art models, there has also been interest in understanding the limitations of these models (given their uninterpretability) (Naik et al., 2018; McCoy et al., 2019), as well as finding systematic biases in benchmark datasets (Gururangan et al., 2018; Poliak et al., 2018).

In parallel to these efforts, there have also been recent logic-based approaches to NLI (Mineshima et al., 2015; Martínez-Gómez et al., 2016; Martínez-Gómez et al., 2017; Abzianidze, 2017; Yanaka et al., 2018), which take inspiration from linguistics. In contrast to early attempts at using logic (Bos and Markert, 2005), these approaches have proven to be more robust. However they tend to use many rules and their output can be hard to interpret. It is sometimes unclear whether the attendant complexity is justified, especially given that such models are currently far outpaced by data-driven models and are generally hard to hybridize with data-driven techniques.

In this work, we introduce a new logical inference engine called MonaLog, which is based on natural logic and work on monotonicity stemming from van Benthem (1986). In contrast to the logical approaches cited above, our starting point is different in that we begin with the following two questions: 1) what is the *simplest* logical system that one can come up with to solve empirical NLI problems (i.e., the system with minimal amounts of primitives and background knowledge)?; and 2) what is the lower-bound performance of such a model? Like other approaches to natural logic (MacCartney and Manning, 2008; Angeli and Manning, 2014), our model works by reasoning over surface forms (as opposed to translating to symbolic representations) using a small inventory of monotonicity facts about quantifiers, lexical items and token-level polarity (Hu and Moss, 2018); *proofs* in the calculus are hence fully interpretable and expressible in ordinary language. Unlike existing work on natural logic, however, our model avoids the need for having expensive alignment and search sub-procedures (MacCartney et al., 2008; Stern and Dagan, 2011), and relies on a much smaller set of background knowledge and primitive relations than MacCartney and Manning (2009).

To show the effectiveness of our approach, we show results on the SICK dataset (Marelli et al., 2014), a common benchmark for logic-based NLI, and find MonaLog to be competitive with more complicated logic-based approaches (many

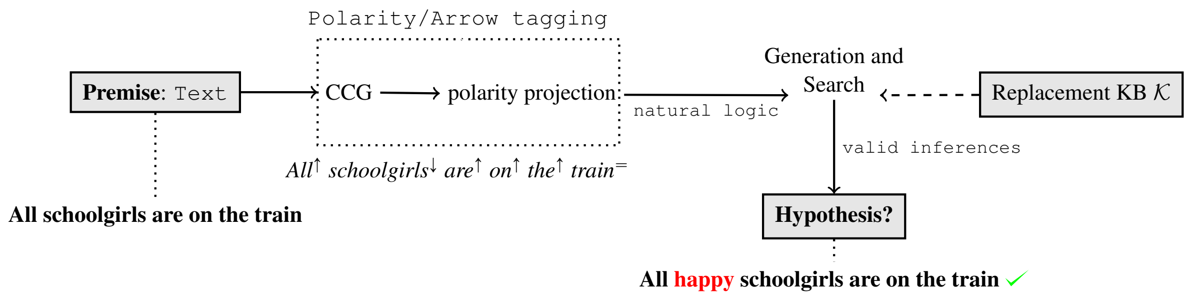


Figure 1: An illustration of our general monotonicity reasoning pipeline using an example premise and hypothesis pair: *All schoolgirls are on the train* and *All happy schoolgirls are on the train*.

of which require full semantic parsing and more complex logical machinery). We also introduce a supplementary version of SICK that corrects several common annotation mistakes (e.g., asymmetrical inference annotations) based on previous work by Kalouli et al. (2017, 2018)[1]. Positive results on both these datasets show the ability of lightweight monotonicity models to handle many of the inferences found in current NLI datasets, hence putting a more reliable lower-bound on what results the simplest logical approach is capable of achieving on this benchmark.

Since our logic operates over surface forms, it is straightforward to hybridize our models. We investigate using MonaLog in combination with the language model BERT (Devlin et al., 2019), including for *compositional data augmentation*, i.e, re-generating entailed versions of examples in our training sets. To our knowledge, our approach is the first attempt to use monotonicity for data augmentation, and we show that such augmentation can generate high-quality training data with which models like BERT can improve performance.

2 Our System: MonaLog

The goal of NLI is to determine, given a premise set P and a hypothesis sentence H, whether H follows from the meaning of P (Dagan et al., 2005). In this paper, we look at single-premise problems that involve making a standard 3-way classification decision (i.e., Entailment (H), Contradict (C) and Neutral (N)). Our general monotonicity reasoning system works according to the pipeline in Figure 1. Given a premise text, we first do `Arrow Tagging` by assigning polarity annotations (i.e., the arrows $\uparrow, \downarrow$, which are the basic primitives of our logic) to tokens in text. These *surface-level* annotations, in turn, are associated with a set of `natural logic` inference rules that provide instructions for how to generate entailments and contradictions by span replacements over these arrows (which relies on a library of span replacement rules). For example, in the sentence *All schoolgirls are on the train*, the token *schoolgirls* is associated with a polarity annotation $\downarrow$, which indicates that in this sentential context, the span *schoolgirls* can be replaced with a semantically more specific concept (e.g., *happy schoolgirls*) in order to generate an entailment. A `generation and search` procedure is then applied to see if the hypothesis text can be generated from the premise using these inference rules. A *proof* in this model is finally a particular sequence of edits (e.g., see Figure 2) that derive the hypothesis text from the premise text rules and yield an entailment or contradiction.

In the following sections, we provide the details of our particular implementation of these different components in MonaLog.

2.1 Polarization (Arrow Tagging)

Given an input premise P, MonaLog first polarizes each of its tokens and constituents, calling the system described by Hu and Moss (2018)[2], which performs polarization on a CCG parse tree. For example, a polarized P could be *every*$^{\uparrow}$ *linguist*$^{\downarrow}$ *swim*$^{\uparrow}$. Note that since we ignore morphology in the system, tokens are represented by lemmas.

2.2 Knowledge Base $\mathcal{K}$ and Sentence Base $\mathcal{S}$

MonaLog utilizes two auxiliary sets. First, a knowledge base $\mathcal{K}$ that stores the world knowledge needed for inference, e.g., *semanticist* $\leqslant$ *linguist* and *swim* $\leqslant$ *move*, which captures the facts that $[\![$*semanticist*$]\!]$ denotes a subset of $[\![$*linguist*$]\!]$,

[1] Our correction can be found at: https://github.com/huhailinguist/SICK_correction

[2] https://github.com/huhailinguist/ccg2mono

and that $[\![swim]\!]$ denotes a subset of $[\![move]\!]$, respectively. Such world knowledge can be created manually for the problem at hand, or derived easily from existing resources such as WordNet (Miller, 1995). Note that we do not blindly add *all* relations from WordNet to our knowledge base, since this would hinge heavily on word sense disambiguation (we need to know whether the "bank" is a financial institution or a river bank to extract its relations correctly). In the current implementation, we avoid this by adding $x \leqslant y$ or $x \perp$[3] y relations only if both x and y are words in the premise-hypothesis pair.[4] Additionally, some relations that involve quantifiers and prepositions need to be hard-coded, since WordNet does not include them: *every* = *all* = *each* ⩽ *most* ⩽ *many* ⩽ *a few* = *several* ⩽ *some* = *a*; *the* ⩽ *some* = *a*; *on* ⊥ *off*; *up* ⊥ *down*; etc.

We also need to keep track of relations that can potentially be derived from the P-H sentence pair. For instance, for all adjectives and nouns that appear in the sentence pair, it is easy to obtain: *adj* + *n* ⩽ *n* (*black cat* ⩽ *cat*). Similarly, we have *n* + *PP/relative clause* ⩽ n (*friend in need* ⩽ *friend*, *dog that bites* ⩽ *dog*), *VP* + *advP/PP* ⩽ *VP* (*dance happily/in the morning* ⩽ *dance*), and so on. We also have rules that extract pieces of knowledge from P directly, e.g.: $n_1 \leqslant n_2$ from sentences of the pattern *every* n_1 *is a* n_2. One can also connect MonaLog to bigger knowledge graphs or ontologies such as DBpedia.

A sentence base $\mathcal{S}$, on the other hand, stores the generated entailments and contradictions.

2.3 Generation

Once we have a polarized CCG tree, and some $\leqslant$ relations in $\mathcal{K}$, generating entailments and contradictions is fairly straightforward. A concrete example is given in Figure 2. Note that the generated $\leqslant$ instances are capable of producing mostly monotonicity inferences, but MonaLog can be extended to include other more complex inferences in *natural logic*, hence the name Mo*na*Log. This extension is addressed in more detail in Hu et al. (2019).

Entailments/inferences The key operation for generating entailments is `replacement`, or substitution. It can be summarized as follows: 1) For upward-entailing (UE) words/constituents, replace them with words/constituents that denote bigger sets. 2) For downward-entailing (DE) words/constituents, either replace them with those denoting smaller sets, or add modifiers (adjectives, adverbs and/or relative clauses) to create a smaller set. Thus for *every*$^{\uparrow}$ *linguist*$^{\downarrow}$ *swim*$^{\uparrow}$, MonaLog can produce the following three entailments by replacing each word with the appropriate word from $\mathcal{K}$: *most*$^{\uparrow}$ *linguist*$^{\downarrow}$ *swim*$^{\uparrow}$, *every*$^{\uparrow}$ *semanticist*$^{\downarrow}$ *swim*$^{\uparrow}$ and *every*$^{\uparrow}$ *linguist*$^{\downarrow}$ *move*$^{\uparrow}$. These are results of one `replacement`. Performing `replacement` for multiple rounds/depths can easily produce many more entailments.

Contradictory sentences To generate sentences contradictory to the input sentence, we do the following: 1) if the sentence starts with "no (some)", replace the first word with "some (no)". 2) If the object is quantified by "a/some/the/every", change the quantifier to "no", and vice versa. 3) Negate the main verb or remove the negation. See examples in Figure 2.

Neutral sentences MonaLog returns Neutral if it cannot find the hypothesis H in $\mathcal{S}.entailments$ or $\mathcal{S}.contradictions$. Thus, there is no need to generate neutral sentences.

2.4 Search

Now that we have a set of inferences and contradictions stored in $\mathcal{S}$, we can simply see if the hypothesis is in either one of the sets by comparing the strings. If yes, then return Entailment or Contradiction; if not, return Neutral, as schematically shown in Figure 2. However, the exact-string-match method is too brittle. Therefore, we apply a heuristic. If the only difference between sentences S_1 and S_2 is in the set {"a", "be", "ing"}, then S_1 and S_2 are considered semantically equivalent.

The search is implemented using depth first search, with a default depth of 2, i.e. at most 2 replacements for each input sentence. At each node, MonaLog "expands" the sentence (i.e., an entailment of its parent) by obtaining its entailments and contradictions, and checks whether H is in either set. If so, the search is terminated; otherwise the systems keeps searching until all the possible entailments and contradictions up to depth 2 have been visited.

[3] ⊥ means "is contradictory to".

[4] There may be better and robust ways of incorporating WordNet relations to $\mathcal{K}$; we leave this for future work.

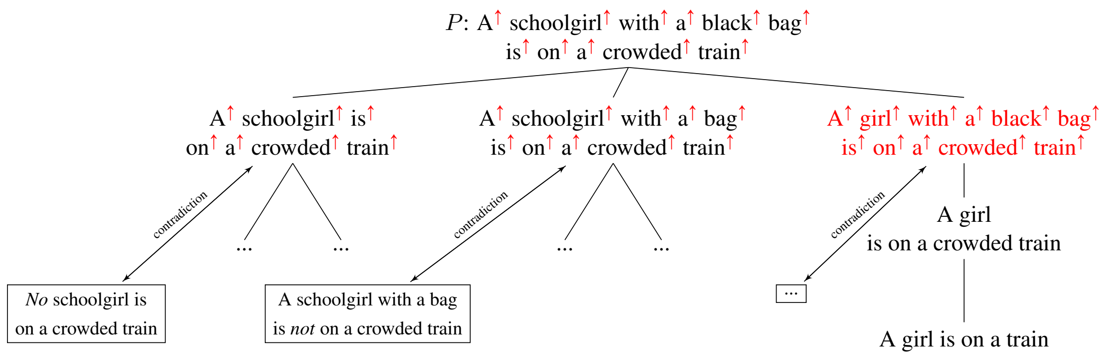


Figure 2: Example search tree for SICK 340, where P is *A schoolgirl with a black bag is on a crowded train*, with the H: *A girl with a black bag is on a crowded train*. Only one `replacement` is allowed at each step. Sentences at the nodes are generated entailments. $\boxed{\text{Sentences}}$ in rectangles are the generated contradictions. In this case our system will return `entail`. The search will terminate after reaching the H in this case, but for illustrative purposes, we show entailments of depth up to 3. To exclude the influence of morphology, all sentences are represented at the lemma level in MonaLog, which is not shown here.

3 MonaLog and SICK

We perform two experiments to test MonaLog. We first use MonaLog to solve the problems in a commonly used natural language inference dataset, SICK (Marelli et al., 2014), comparing our results with previous systems. Second, we test the quality of the data generated by MonaLog. To do this, we generate more training data (sentence pairs) from the SICK training data using our system, and performe fine-tuning on BERT (Devlin et al., 2019), a language model based on the transformer architecture (Vaswani et al., 2017), with the expanded dataset. In all experiments, we use the Base, Uncased model of BERT[5].

3.1 The SICK Dataset

The SICK (Marelli et al., 2014) dataset includes around 10,000 English sentence pairs that are annotated to have either "Entailment", "Neutral" or "Contradictory" relations. We choose SICK as our testing ground for several reasons. First, we want to test on a large-scale dataset, since we have shown that a similar model (Hu et al., 2019) reaches good results on parts of the smaller FraCaS dataset (Cooper et al., 1996). Second, we want to make our results comparable to those of previous logic-based models such as the ones described in (Bjerva et al., 2014; Abzianidze, 2015; Martínez-Gómez et al., 2017; Yanaka et al., 2018), which were also tested on SICK. We use the data split provided in the dataset: 4,439 training problems, 4,906 test problems and 495 trial problems, see Table 1 for examples.

3.2 Hand-corrected SICK

There are numerous issues with the original SICK dataset, as illustrated by Kalouli et al. (2017, 2018).

They first manually checked 1,513 pairs tagged as "A entails B but B is neutral to A" (*AeBBnA*) in the original SICK, correcting 178 pairs that they considered to be wrong (Kalouli et al., 2017). Later, Kalouli et al. (2018) extracted pairs from SICK whose premise and hypothesis differ in only one word, and created a simple rule-based system that used WordNet information to solve the problem. Their WordNet-based method was able to solve 1,651 problems, whose original labels in SICK were then manually checked and corrected against their system's output. They concluded that 336 problems are wrongly labeled in the original SICK. Combining the above two corrected subsets of SICK, minus the overlap, results in their corrected SICK dataset[6], which has 3,016 problems (3/10 of the full SICK), with 409 labels different from the original SICK (see breakdown in Table 2). 16 of the corrections are in the trial set, 197 of them in the training set and 196 in the test set. This suggests that more than one out of ten problems in SICK are potentially problematic. For this reason, two authors of the current paper checked the 409 changes. We found that only 246 problems are labeled the same by our team and by Kalouli et al. (2018). For cases where there is disagreement, we adjudicated the differences after a

[5] https://github.com/google-research/bert

[6] https://github.com/kkalouli/SICK-processing

id	premise	hypothesis	orig. label	corr. label
219	There is no girl in white dancing	A girl in white is dancing	C	C
294	Two girls are lying on the ground	Two girls are sitting on the ground	N	C
743	A couple who have just got married are walking down the isle	The bride and the groom are leaving after the wedding	E	N
1645	A girl is on a jumping car	One girl is jumping on the car	E	N
1981	A truck is quickly going down a hill	A truck is quickly going up a hill	N	C
8399	A man is playing guitar next to a drummer	A guitar is being played by a man next to a drummer	E	n.a.

Table 1: Examples from SICK (Marelli et al., 2014) and corrected SICK (Kalouli et al., 2017, 2018) w/ syntactic variations. n.a.:example not checked by Kalouli and her colleagues. C: contradiction; E: entailment; N: neutral.

total	N → E	E → C	N → C	E → N
409	14	7	190	198

Table 2: Changes from SICK to corrected SICK (Kalouli et al., 2017, 2018).

discussion.

We are aware that the partially checked SICK (by two teams) is far from ideal. We therefore present results for two versions of SICK for experiment 1 (section 4): the original SICK and the version corrected by our team. For the data augmentation experiment in section 5, we only performed fine-tuning on the corrected SICK. As shown in a recent SICK annotation experiment by Kalouli et al. (2019), annotation is a complicated issue influenced by linguistic and non-linguistic factors. We leave checking the full SICK dataset to future work.

4 Experiment 1: Using MonaLog Directly

4.1 Setup and Preprocessing

The goal of experiment 1 is to test how accurately MonaLog solves problems in a large-scale dataset. We first used the system to solve the 495 problems in the trial set and then manually identified the cases in which the system failed. Then we determined which syntactic transformations are needed for MonaLog. After improving the results on the trial data by introducing a preprocessing step to handle limited syntactic variation (see below), we applied MonaLog on the test set. This means that the rule base of the system was optimized on the trial data, and we can test its generalization capability on the test data.

The main obstacle for MonaLog is the syntactic variations in the dataset, illustrated in some examples in Table 1. There exist multiple ways of dealing with these variations: One approach is to 'normalize' unknown syntactic structures to a known structure. For example, we can transform passive sentences into active ones and convert existential sentences into the base form (see ex. 8399 and 219 in Table 1). Another approach is to use some more abstract syntactic/semantic representation so that the linear word order can largely be ignored, e.g., represent a sentence by its dependency parse, or use Abstract Meaning Representation. Here, we explore the first option and leave the second approach to future work. We believe that dealing with a wide range of syntactic variations requires tools designed specifically for that purpose. The goal of MonaLog is to generate entailments and contradictions based on a polarized sentence instead.

Below, we list the most important syntactic transformations we perform in preprocessing[7].

1. Convert all passive sentences to active using *pass2act*[8]. If the passive does not contain a *by* phrase, we add *by a person*.
2. Convert existential clauses into their base form (see ex. 219 in Table 1).
3. Other transformations: *someone/anyone/no one* → *some/any/no person*; *there is no man doing sth.* → *no man is doing sth.*; etc.

4.2 Results

The results of our system on uncorrected and corrected SICK are presented in Table 3, along with comparisons with other systems.

[7] For the complete list of transformations see: https://github.com/huhailinguist/SICK_correction

[8] https://github.com/DanManN/pass2act

system	P	R	acc.
On **uncorrected** SICK			
majority baseline	–	–	56.36
hypothesis-only baseline (Poliak et al., 2018)	–	–	56.87
MonaLog (this work)			
MonaLog + all transformations	83.75	70.66	77.19
Hybrid: MonaLog + BERT	83.09	85.46	85.38
ML/DL-based systems			
BERT (base, uncased)	86.81	85.37	86.74
(Yin and Schütze, 2017)	–	–	**87.1**
(Beltagy et al., 2016)	–	–	85.1
Logic-based systems			
(Bjerva et al., 2014)	93.6	60.6	81.6
(Abzianidze, 2015)	97.95	58.11	81.35
(Martínez-Gómez et al., 2017)	97.04	63.64	83.13
(Yanaka et al., 2018)	84.2	77.3	84.3
On **corrected** SICK			
MonaLog + existential trans.	89.43	71.53	79.11
MonaLog + pass2act	89.42	72.18	80.25
MonaLog + all transformations	89.91	74.23	81.66
Hybrid: MonaLog + BERT	85.65	87.33	**85.95**
BERT (base, uncased)	84.62	84.27	85.00

Table 3: Performance on the SICK test set, original SICK above and corrected SICK below. P / R for MonaLog averaged across three labels. Results involving BERT are averaged across six runs; same for later experiments.

Our accuracy on the uncorrected SICK (77.19%) is much higher than the majority baseline (56.36%) or the hypothesis-only baseline (56.87%) reported by Poliak et al. (2018), and only several points lower than current logic-based systems. Since our system is based on *natural logic*, there is no need for translation into logical forms, which makes the reasoning steps transparent and much easier to interpret. I.e., with entailments and contradictions, we can generate a natural language trace of the system, see Fig. 2.

Our results on the corrected SICK are even higher (see lower part of Table 3), demonstrating the effect of data quality on the final results. Note that with some simple syntactic transformations we can gain 1-2 points in accuracy.

Table 4 shows MonaLog's performance on the individual relations. The system is clearly very good at identifying entailments and contradictions, as demonstrated by the high precision values, especially on the corrected SICK set (98.50 precision for E and 95.02 precision for C). The lower recall values are due to MonaLog's current inability to handle syntactic variation.

Based on these results, we tested a hybrid model of MonaLog and BERT (see Table 3) where we exploit MonaLog's strength: Since MonaLog has a very high precision on Entailment and Contradiction, we can always trust MonaLog if it predicts E or C; when it returns N, we then fall back to BERT. This hybrid model improves the accuracy of BERT by 1% absolute to 85.95% on the corrected SICK. On the uncorrected SICK dataset, the hybrid system performs worse than BERT. Since MonaLog is optimized for the corrected SICK, it may mislabel many E and C judgments in the *uncorrected* dataset. The stand-alone BERT system performs better on the uncorrected data (86.74%) than the corrected set (85.00%). The corrected set may be too inconsistent since only a part has been checked.

Overall, these hybird results show that it is possible to combine our high-precision system with deep learning architectures. However, more work is necessary to optimize this combined system.

4.3 Error Analysis

Upon closer inspection, some of MonaLog's errors consist of difficult cases, as shown in Table 5. For example, in ex. 359, if our knowledge base $\mathcal{K}$ contains the background fact *chasing* $\leqslant$ *running*, then MonaLog's judgment of C would be correct. In ex. 1402, if *crying* means *screaming*, then the label should be E; however, if *crying* here means *shedding tears*, then the label should probably be N. Here we also see potentially problematic labels (ex. 1760, 3403) in the original SICK dataset.

Another point of interest is that 19 of MonaLog's mistakes are related to the antonym pair *man* vs. *woman* (e.g., ex. 5793 in Table 5). This points to inconsistency of the SICK dataset: Whereas there are at least 19 cases tagged as Neutral (e.g., ex. 5793), there are at least 17 such pairs that are annotated as Contradictions in the test set (e.g., ex. 3521), P: *A man is dancing*, H: *A woman is dancing* (ex. 9214), P: *A shirtless man is jumping over a log*, H: *A shirtless woman is jumping over a log*. If *man* and *woman* refer to the same entity, then clearly that entity cannot be *man* and *woman* at the same time, which makes the sentence pair a contradiction. If, however, they do not refer to the same entity, then they should be Neutral.

	E		C		N	
	P	R	P	R	P	R
uncorr. SICK	97.75	46.74	80.06	70.24	73.43	94.99
corr. SICK	98.50	50.46	95.02	73.60	76.22	98.63

Table 4: Results of MonaLog per relation. C: contradiction; E: entailment; N: neutral.

id	premise	hypothesis	SICK	corr. SICK	Mona
359	There is no dog chasing another or holding a stick in its mouth	Two dogs are running and carrying an object in their mouths	N	n.a.	C
1402	A man is crying	A man is screaming	N	n.a.	E
1760	A flute is being played by a girl	There is no woman playing a flute	N	n.a.	C
2897	The man is lifting weights	The man is lowering barbells	N	n.a.	E
2922	A herd of caribous is not crossing a road	A herd of deer is crossing a street	N	n.a.	C
3403	A man is folding a tortilla	A man is unfolding a tortilla	N	n.a.	C
4333	A woman is picking a can	A woman is taking a can	E	N	E
5138	A man is doing a card trick	A man is doing a magic trick	N	n.a.	E
5793	A man is cutting a fish	A woman is slicing a fish	N	n.a.	C

Table 5: Examples of incorrect answers by MonaLog; n.a. = the problem has not been checked in corr. SICK.

5 Experiment 2: Data Generation Using MonaLog

Our second experiment focuses on using MonaLog to generate additional training data for machine learning models such as BERT. To our knowledge, this is the first time that a rule-based NLI system has been successfully used to generate training data for a deep learning application.

5.1 Setup

As described above, MonaLog generates entailments and contradictions when solving problems. These can be used as additional training data for a machine learning model. I.e., we pair the newly generated sentences with their input sentence, creating new pairs for training. For example, we take all the sentences in the *nodes* in Figure 2 as inferences and all the sentences in *rectangles* as contradictions, and then form sentence pairs with the input sentence. The additional data can be used directly, almost without human intervention.

Thus for experiment 2, the goal is to examine the quality of these generated sentence pairs. For this, we re-train a BERT model on these pairs. If BERT trained on the manually annotated SICK training data is improved by adding data generated by MonaLog, then we can conclude that the generated data is of high quality, even comparable to human annotated data, which is what we found.

More specifically, we compare the performance of BERT models trained on a) SICK training data alone, and b) SICK training data plus the entailing and contradictory pairs generated by MonaLog. All experiments are carried out using our corrected version of the SICK data set.

However, note that MonaLog is designed to only generate entailments and contradictions. Thus, we only have access to newly generated examples for those two cases, we do not acquire any additional neutral cases. Consequently, adding these examples to the training data will introduce a skewing that does not reflect the class distribution in the test set. Since this will bias the machine learner against neutral cases, we use the following strategy to counteract that tendency: We relabel all cases where BERT is not confident enough for either E or C into N. We set this threshold to 0.95 but leave further optimization of the threshold to future work.

5.2 Data Filtering and Quality Control

MonaLog is prone to over-generation. For example, it may wrongly add the same adjective before a noun (phrase) twice to create a more specific noun, e.g., *young young man* $\leqslant$ *young man* $\leqslant$

label	premise	hypothesis	comm.
E	A woman be not cooking something	A person be not cooking something	correct
E	A man be talk to a woman who be seat beside he and be drive a car	A man be talk	correct
E	A south African plane be not fly in a blue sky	A south African plane be not fly in a very blue sky in a blue sky	unnat.
C	No panda be climb	Some panda be climb	correct
C	A man on stage be sing into a microphone	A man be not sing into a microphone	correct
C	No man rapidly be chop some mushroom with a knife	Some man rapidly be chop some mushroom with a knife with a knife	unnat.
E	Few$^{\uparrow}$ people$^{\downarrow}$ be$^{\downarrow}$ eat$^{\downarrow}$ at$^{\downarrow}$ red$^{\downarrow}$ table$^{\downarrow}$ in$^{\downarrow}$ a$^{\downarrow}$ restaurant$^{\downarrow}$ without$^{\downarrow}$ light$^{\uparrow}$	Few$^{\uparrow}$ large$^{\downarrow}$ people$^{\downarrow}$ be$^{\downarrow}$ eat$^{\downarrow}$ at$^{\downarrow}$ red$^{\downarrow}$ table$^{\downarrow}$ in$^{\downarrow}$ a$^{\downarrow}$ Asian$^{\downarrow}$ restaurant$^{\downarrow}$ without$^{\downarrow}$ light$^{\uparrow}$	correct

Table 6: Sentence pairs generated by MonaLog, lemmatized.

label	total	correct	wrong	unnatural
E	56	49	0	7
C	44	41	0	3

Table 7: Quality of 100 manually inspected sentences.

man. Since it is possible that such examples influence the machine learning model negatively, we look into filtering such examples to improve the quality of the additional training data.

We manually inspected 100 sentence pairs generated by MonaLog to check the quality and naturalness of the new sentences (see Table 6 for examples). All of the generated sentences are correct in the sense that the relation between the premise and the hypothesis is correctly labeled as entailment or contradiction (see Table 7). While we did not find any sentence pairs with wrong labels, some generated sentences are unnatural, as shown in Table 6. Both unnatural examples contain two successive copies of the same PP.

Note that our data generation hinges on correct polarities on the words and constituents. For instance, in the last example of Table 6, the polarization system needs to know that *few* is downward entailing on both of its arguments, and *without* flips the arrow of its argument, in order to produce the correct polarities, on which the `replacement` of MonaLog depends.

In order to filter unnatural sentences, such as the examples in Table 6, we use a rule-based filter and remove sentences that contain bigrams of repeated words[9]. We experiment with using one quarter or one half randomly selected sentences in addition to a setting where we use the complete set of generated sentences.

5.3 Results

Table 8 shows the amount of additional sentence pairs per category along with the results of using the automatically generated sentences as additional training data.

It is obvious that adding the additional training data results in gains in accuracy even though the training data becomes increasingly skewed towards E and C. When we add all additional sentence pairs, accuracy increases by more than 1.5 percent points. This demonstrates both the robustness of BERT in the current experiment and the usefulness of the generated data. The more data we add, the better the system performs.

We also see that raising the threshold to relabel uncertain cases as neutral gives a small boost, from 86.51% to 86.71%. This translates into 10 cases where the relabeling corrected the answer.

Finally, we also investigated whether the hybrid system, i.e., MonaLog followed by the retrained BERT, can also profit from the additional training data. Intuitively, we would expect smaller gains since MonaLog already handles a fair amount of the entailments and contradictions, i.e., those cases where BERT profits from more examples. However the experiments show that the hybrid system reaches an even higher accuracy of 87.16%, more than 2 percent points above the

[9] We also investigated using a bigram based language model to filter out non-natural sentences. However, this affected the results negatively.

training data	# E	# N	# C	acc.
SICK.train: baseline	1.2k	2.5k	0.7k	85.00
1/4 gen. + SICK.train	8k	2.5k	4k	85.30
1/2 gen. + SICK.train	15k	2.5k	7k	85.81
all gen. + SICK.train	30k	2.5k	14k	86.51
E, C prob. threshold = 0.95	30k	2.5k	14k	86.71
Hybrid baseline	1.2k	2.5k	0.7k	85.95
Hybrid + all gen.	30k	2.5k	14k	87.16
Hybrid + all gen. + threshold	30k	2.5k	14k	**87.49**

Table 8: Results of BERT trained on MonaLog-generated entailments and contradictions plus SICK.train (using the corrected SICK set).

baseline, equivalent to roughly 100 more problems correctly solved. Setting the high threshold for BERT to return E or C further improves accuracy to 87.49%. This brings us into the range of the state-of-the-art results, even though a direct comparison is not possible because of the differences between the corrected and uncorrected dataset.

6 Conclusions and Future Work

We have presented a working natural-logic-based system, MonaLog, which attains high accuracy on the SICK dataset and can be used to generated natural logic proofs. Considering how simple and straightforward our method is, we believe it can serve as a strong baseline or basis for other (much) more complicated systems, either logic-based or ML/DL-based. In addiction, we have shown that MonaLog can generate high-quality training data, which improves the accuracy of a deep learning model when trained on the expanded dataset. As a minor point, we manually checked the corrected SICK dataset by Kalouli et al. (2017, 2018).

There are several directions for future work. The first direction concerns the question how to handle syntactic variation from natural language input. That is, the computational process(es) for inference will usually be specified in terms of strict syntactic conditions, and naturally occurring sentences will typically not conform to those conditions. Among the strategies which allow their systems to better cope with premises and hypotheses with various syntactic structures are sophisticated versions of alignment used by e.g. MacCartney (2009); Yanaka et al. (2018). We will need to extend MonaLog to be able to handle such variation. In the future, we plan to use dependency relations as representations of natural language input and train a classifier that can determine which relations are crucial for inference.

Second, as mentioned earlier, we are in need of a fully (rather than partially) checked SICK dataset to examine the impact of data quality on the results since the partially checked dataset may be inherently inconsistent between the checked and non-checked parts.

Finally, with regard to the machine learning experiments, we plan to investigate other methods of addressing the imbalance in the training set created by additional entailments and contradictions. We will look into options for artificially creating neutral examples, e.g. by finding reverse entailments[10], as illustrated by Richardson et al. (2019).

Acknowledgements

We thank the anonymous reviewers for their helpful comments. Hai Hu is supported by China Scholarship Council.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of EMNLP*, pages 2492–2502.

Lasha Abzianidze. 2017. LangPro: Natural language theorem prover. In *Proceedings of EMNLP: System Demonstrations*, pages 115–120, Copenhagen, Denmark.

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of EMNLP*, pages 534–545, Doha, Qatar.

Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J Mooney. 2016. Representing meaning with a combination of logical and

[10] In the set relations by MacCartney (2009), if $A \sqsubset B$, then A entails B, but B is neutral to A.

distributional models. *Computational Linguistics*, 42(4):763–808.

Johan van Benthem. 1986. *Essays in Logical Semantics*, volume 29 of *Studies in Linguistics and Philosophy*. D. Reidel Publishing Co., Dordrecht.

Johannes Bjerva, Johan Bos, Rob Van der Goot, and Malvina Nissim. 2014. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646.

Johan Bos and Katja Markert. 2005. Recognising Textual Entailment with Logical Inference. In *Proceedings of EMNLP*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 177–190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of NAACL: HLT*, volume 2, pages 107–112.

Hai Hu, Qi Chen, and Lawrence S Moss. 2019. Natural language inference with monotonicity. In *Proceedings of the 13th International Conference on Computational Semantics (IWCS)*.

Hai Hu and Lawrence S. Moss. 2018. Polarity computations in flexible categorial grammar. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 124–129.

Aikaterini-Lida Kalouli, Annebeth Buis, Livy Real, Martha Palmer, and Valeria dePaiva. 2019. Explaining simple natural language inference. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 132–143.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2017. Textual inference: Getting logic from humans. In *IWCS 2017: 12th International Conference on Computational Semantics*.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2018. Wordnet for easy textual inferences. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.

Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP*, pages 802–811. Association for Computational Linguistics.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of COLING*, pages 521–528.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *IWCS-8, Proceedings of the Eighth International Conference on Computational Semantics*, pages 140–156.

M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL 2016 System Demonstrations*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of EACL*, pages 710–720.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *arXiv preprint arXiv:1902.01007*.

George A Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of EMNLP*, pages 2055–2061.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Kyle Richardson, Hai Hu, Lawrence S Moss, and Ashish Sabharwal. 2019. Probing natural language inference models through semantic fragments. *arXiv preprint arXiv:1909.07521*.

Asher Stern and Ido Dagan. 2011. A confidence model for syntactically-motivated entailment proofs. In *Proceedings of RANLP*, pages 455–462.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, volume 1, pages 1112–1122.

Hitomi Yanaka, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki. 2018. Acquisition of phrase correspondences using natural deduction proofs. In *Proceedings of NAACL-HLT*, pages 756–766, New Orleans, LA.

Wenpeng Yin and Hinrich Schütze. 2017. Task-specific attentive pooling of phrase alignments contributes to sentence matching. In *Proceedings of EACL*, pages 699–709.

Tier-Based Strictly Local Stringsets: Perspectives from Model and Automata Theory

Dakotah Lambert
Stony Brook University
Department of Linguistics
dakotah.lambert@stonybrook.edu

James Rogers
Earlham College
Department of Computer Science
jrogers@cs.earlham.edu

Abstract

Defined by Heinz et al. (2011) the Tier-Based Strictly Local (TSL) class of stringsets has not previously been characterized by an abstract property that allows one to prove a stringset's membership or lack thereof. We provide here two such characterizations: a generalization of suffix substitution closure and an algorithm based on deterministic finite-state automata (DFAs). We use the former to prove closure properties of the class. Additionally, we extend the approximation and constraint-extraction algorithms of Rogers and Lambert (2019a) to account for TSL constraints, allowing for free conversion between TSL logical formulae and DFAs.

1 Tier-Based Strict Locality

The class of Strictly k-Local stringsets (SL_k), first described by McNaughton and Papert (1971), is well known, with learning algorithms from Garcia et al. (1990) and a decision algorithm stemming from Caron (1998) that led to constraint-extraction algorithms from Rogers and Lambert (2019a). A superclass of this, defined by the application of a Strictly k-Local grammar to the output of an erasing homomorphism (which may be the identity map) was introduced by Heinz et al. (2011) as the Tier-Based Strictly k-Local sets of strings.

In this paper, we introduce a purely relational view of TSL. From this, we derive a generalization of the abstract characterization of the Strictly k-Local stringsets for their tier-based cousins, extend the known approximation and constraint-extraction algorithms to this class, and introduce a type of alphabet-agnostic finite-state automaton, and operations thereon, useful in building representations of stringsets from logical formulae.

In demonstration of the abstract characterization of the class, we prove that TSL is not, in general, closed under any of the Boolean operations. We demonstrate in contrast that intersection closure does hold when the tier alphabets are the same. We then investigate and classify some specific linguistic examples, namely the one-stress constraint, the liquid dissimilation of Latin, and the backness harmony of Uyghur.

2 Relational Word Models

We begin by defining a relational word model in the same way as Rogers and Lambert (2019b). A relational structure in general is a set of domain elements, D, augmented with a set of relations of arbitrary arity, $R_i \subseteq D^{n_i}$. Let w be a string over some alphabet Σ. Then a *word model* for w is a structure:

$$\mathcal{M}_{\Sigma}^{R_i}(w) \triangleq \langle D_w, \sigma_w, \rtimes_w, \ltimes_w, R_i \rangle_{\sigma \in \Sigma}.$$

where D_w is isomorphic to an initial segment $\langle 0, 1, \ldots, |w|+1 \rangle$ of the natural numbers and represents the positions in $\rtimes w \ltimes$, each σ_w (in addition to $\rtimes_w$ and $\ltimes_w$) is a unary relation that holds for all and only those positions at which σ (or $\rtimes$ or $\ltimes$, respectively) occurs, and the remaining R_i are the other salient relations, such as the standard successor or precedence relations (denoted in this paper by $\lhd$ and $<$, respectively). Note that the set $\{\sigma_w, \rtimes_w, \ltimes_w\}_{\sigma \in \Sigma}$ is a partition of D_w. As a minor abuse of notation, we allow symbols to refer to their associated relations, and we allow sets of relations of the same arity to be read as the disjunction of their pointwise application. Figure 1 shows three different word models for the string "abab", where each cell represents a domain element, each cell's label is the alphabetic unary relation that element satisfies, and the edges represent the indicated relation. The tier-successor relation, $\lhd^{\tau}$, will be defined shortly hereafter.

The class of Strictly k-Local stringsets over a tier $\tau \subseteq \Sigma$, TSL_k^{τ}, was originally described as

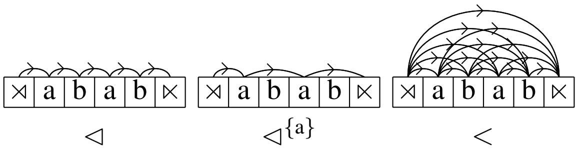


Figure 1: Three word models for the string "abab", the first variant under the standard successor relation, the second under the tier-successor relation on the alphabet $\{a\}$, and the last under the precedence relation.

those characterized by a set of Strictly k-Local constraints on the output of an erasing homomorphism (Heinz et al., 2011). Note that it can be assumed that the tier alphabet always contains $\rtimes$ and $\ltimes$. Here we suggest an alternative perspective based on relational word models and define a relation appropriate for describing this class.

The standard successor relation is the transitive reduction of the precedence relation and is first-order definable from the latter as follows:

$$x \lhd y \triangleq (x < y) \wedge (\forall z)\big[\neg(x \leq z \leq y)\big].$$

With minor modification, we can instead use the restriction of the precedence relation to the intended tier-alphabet and derive a similar relation:

$$\begin{aligned} x \lhd^{\tau} y \triangleq{} & T(x) \wedge T(y) \wedge (x < y) \\ & \wedge (\forall z)\big[\neg\big(T(z) \wedge (x \leq z \leq y)\big)\big]. \end{aligned}$$

This definition is equivalent to the standard successor relation after erasing symbols not in the intended tier alphabet, and through this equivalence we use this tier-successor relation as our basis for describing TSL stringsets and constraints. By extension, this relation should be useful in describing the yet unexplored Boolean closure of TSL^{τ} formulae, which we call Tier-Based k-Locally Testable analogously to the Locally Testable and Piecewise Testable classes characterized by McNaughton and Papert (1971) and Simon (1975), respectively. We will revisit this in section 8.

In order to avoid doubling sub- and superscripts, the tier-successor relation over tier τ is written $\lhd[\tau]$ when it appears in such a position.

3 Windows and Factors

Given a homogeneous relation R of arity a, the set

$$W_a^R(x_1) \triangleq \{x_1 \ldots x_a : \langle x_1, \ldots, x_a\rangle \in R\}$$

is the set of windows of length a (a-windows) that begin with x_1. The set of windows of length $n > a$ is defined inductively:

$$\begin{aligned} W_{i+1}^R(x_1) \triangleq \{ & x_1 \ldots x_{i+1} : \\ & x_1 \ldots x_i \in W_i^R(x_1) \text{ and} \\ & \langle x_{i-a+2}, \ldots, x_{i+1}\rangle \in R\}. \end{aligned}$$

Informally, each n-window is a sequence of positions that can be formed from a sequence of overlapping a-windows, the latter being sequences formed directly from the tuples in R. In order to discuss windows shorter than the arity of their defining relation, we say that any of the affixes of an n-window of length $m < n$ is an m-window from an appropriate starting point. Let the first position of a string x be denoted by p_0 and the final one by p_f, then define the length of x under the relation R as the size of the largest window that can be formed in x:

$$|x|^R \triangleq \max\Big\{n : (\exists v)\big[vp_f \in W_n^R(p_0)\big]\Big\}.$$

If R is a binary relation for which the transitive closure is asymmetric, such as the $<$ relation or its reductions used in this paper, $|x|^R$ is finite whenever x is itself finite.

Let $\hat{\Sigma} = \Sigma \cup \{\rtimes, \ltimes\}$. A string $s = \hat{\sigma}_1\hat{\sigma}_2 \ldots \hat{\sigma}_k$ for $\hat{\sigma}_i \in \hat{\Sigma}$ is a *k-factor* of a string t under the relation R, $s \sqsubseteq^R t$, iff for some position $p \in D_t$ there is some k-window $w_1 w_2 \ldots w_k \in W_k^R(p)$ such that each $\hat{\sigma}_i$ holds for the corresponding w_i. For example, one can use Figure 1 to see that for both the $\lhd^{\{a\}}$ and $<$ relations, it holds that aa $\sqsubseteq$ ⋊abab⋉, but not for $\lhd$. Additionally, abb $\sqsubseteq$ ⋊abab⋉ for $<$ but neither for $\lhd$ nor for $\lhd^{\{a\}}$.

Define the set of all k-factors of w as follows:

$$F_k^R(w) \triangleq \big\{s : |s| = k \text{ and } s \sqsubseteq^R w\big\}.$$

Additionally, define the set of factors of width at most k as one would expect:

$$F_{\leq k}^R(w) \triangleq \bigcup_{1 \leq i \leq k} \big(F_i^R(w)\big).$$

Note that a window is distinct from a factor in that the former is a sequence of positions while the latter describes a string of symbols that occupies such a sequence of positions.

Following Rogers and Lambert (2019b), we say a function $f\colon X^n \to X$ is *conservative* iff f preserves well-formedness of its inputs and it holds that for all possible inputs:

$$F_k^R\big(f(x_1, \ldots, x_n)\big) \subseteq \bigcup_{1 \leq i \leq n} \big(F_k^R(x_i)\big).$$

For strings inserting and deleting symbols other than end-markers preserves well-formedness. Note that conservativity of an operation depends on R, k, and the domain; for example, while inserting or deleting symbols not in τ is conservative under $\triangleleft^{\tau}$ (since $\triangleleft^{\tau}$ ignores them), the insertion is not conservative under $<$.

A factor f may be taken as a logical proposition that f occurs. A word model $\mathcal{M}(w)$ satisfies such a proposition, $\mathcal{M}(w) \models f$, iff $f \sqsubseteq w$. Satisfaction of a set of factors is considered disjunctively, and the Boolean connectives hold their usual meaning.

If φ is an arbitrary logical sentence using these constructions, the *models* of φ are the structures:

$$\mathrm{Mod}(\varphi) \triangleq \{\mathcal{M} : \mathcal{M} \models \varphi\},$$

and one can say that φ represents the stringset:

$$L(\varphi) \triangleq \{w : \mathcal{M}(w) \in \mathrm{Mod}(\varphi)\}.$$

Any stringset definable in this way is said to be *locally definable* under the relations in question, as an extension of the notion of locality used by McNaughton and Papert (1971). A logic further restricted to φ of the form:

$$\varphi = \bigwedge(\neg f_i)$$

where each f_i is a factor (a conjunction of negative literals) characterizes those stringsets that are *locally definable in the strict sense*.

The Strictly k-Local stringsets and their tier-based cousins are definable by a set of permitted k-factors over the appropriate relation $G \subseteq \hat{\Sigma}^{\leq k}$. We call such G a *grammar*. Since for a finite alphabet there are only finitely many k-factors, we could equivalently use the complement of G, denoted $\overline{G}$. Then the stringset is locally definable in the strict sense by taking $\varphi = \bigwedge(\neg f \in \overline{G})$.

Any stringset locally definable in the strict sense is closed under any operation conservative under the appropriate relations and factor width, because if no factor of any input is forbidden and the operation does not introduce new factors, the output cannot contain a forbidden factor.

4 Substitution of (Preprojective) Suffixes

A property is said to *characterize* a class iff all members of the class have the property and all objects that have the property are members of the class. For example, the Strictly k-Local stringsets are characterized by closure under substitution of suffixes (Rogers and Pullum, 2011). When two strings in an SL_k set share a factor of width $k-1$, the portions following this shared factor in each may be swapped to obtain new strings in the set. In order to describe an analogous property for TSL, first define the *projection* of w onto τ as follows:

$$\pi_{\tau}(w) \triangleq F^{\triangleleft[\tau]}_{|w|^{\triangleleft[\tau]}}(w).$$

In other words, $\pi_{\tau}(w)$ is the set of $\triangleleft^{\tau}$ factors in w the same length as the longest such factor. It can be shown that this is singleton and equivalent to the standard projection operation. We omit tier specifications when they are clear from context. Following mathematical tradition, we abuse notation and use $\pi_{\tau}(w)$ to refer to its single element.

To move freely between strings and projections, we note the following:

Lemma 1. *If a stringset L over some alphabet Σ is closed under insertion and deletion of symbols outside of some $\tau \subseteq \Sigma$, then $w \in L$ iff $\pi_{\tau}(w) \in L$.*

Proof. Let L be so closed. If w in L, then by closure under deletion, $\pi_{\tau}(w) \in L$. If $\pi_{\tau}(w) \in L$, then by closure under insertion, $w \in L$. □

Definition 1 (Preprojective Suffix Substitution)**.** Let Σ be an alphabet and $\tau \subseteq \Sigma$ a tier-alphabet. Let $w_1 = u_1x_1v_1$ and $w_2 = u_2x_2v_2$ be strings over Σ^* such that $\pi_{\tau}(x_1) = \pi_{\tau}(x_2)$. We then say the substrings x_1 and x_2 are *projectively shared factors* of size $k = |x_1|^{\triangleleft[\tau]}$ and the string $w_3 = u_1x_1v_2$ is formed by *τ-preprojective suffix substitution.*

For strings on τ^*, preprojective suffix substitution is identical to the standard suffix substitution under which SL stringsets are closed. Further, recall that insertion and deletion of symbols outside of τ is conservative, and so TSL stringsets are closed under these operations. Preprojective suffix substitution is equivalent to projecting onto τ, performing suffix substitution on the restricted domain, then doing an inverse projection by reinserting the symbols that were removed earlier. Since each step is conservative, preprojective suffix substitution is as well, so TSL stringsets are closed thereunder. More interesting is the following:

Theorem 1. *All stringsets closed both under insertion and deletion of symbols outside of some tier alphabet τ and under τ-preprojective suffix substitution for some factor size k are* TSL^{τ}.

Proof. Let L be a stringset so closed. Since L is closed under τ-preprojective suffix substitution, its projection to τ ($\pi_\tau(L)$) is closed under suffix substitution and is thus SL_k. Further, for any $w \in \Sigma^*$ such that $\pi_\tau(w)$ is in $\pi_\tau(L)$, Lemma 1 guarantees that w is itself in L (and vice versa). Thus by definition, L is TSL_k^τ. □

Since all TSL stringsets are closed under these operations and all stringsets so closed are TSL, this combination of closures characterizes TSL.

5 Closure Properties

One constraint that is nearly universal in phonotactics is that one and only one syllable with primary stress ($\acute{\sigma}$) occurs in a given word (Hyman, 2009). Despite the fact that this constraint as a whole is neither Strictly Local nor Strictly Piecewise, it is $\mathrm{TSL}_2^{\{\acute{\sigma}\}}$, as witnessed by the following formula:

$$\neg \rtimes\ltimes \wedge \neg\acute{\sigma}\acute{\sigma}.$$

While similar formulae show that TSL_{n+1}^τ can require that n instances of arbitrary elements from τ occur, we can prove, for example, that no TSL stringset can recognize exactly the set of strings containing both 'a' and 'b'. Since TSL is closed under deletion of non-tier symbols and "ab" is in L but neither "a" nor "b" is itself in L, it is necessarily the case that both symbols would have to be on the tier alphabet for any TSL grammar that recognizes L. Using strings formed from these symbols alone, we can demonstrate failure of preprojective suffix substitution closure for TSL_3:

$$\begin{aligned} w_1 &= \rtimes\boxed{\mathrm{aa}}\mathrm{b}\ltimes \in \rtimes L\ltimes \\ w_2 &= \rtimes\mathrm{b}\boxed{\mathrm{aa}}\ltimes \in \rtimes L\ltimes \\ w_3 &= \rtimes\boxed{\mathrm{aa}}\ltimes \notin \rtimes L\ltimes. \end{aligned}$$

In fact, by making the shared 'a' factor be of width $k-1$ rather than 2, it can be shown that no TSL_k grammar can describe exactly the set of strings containing both 'a' and 'b'. This is despite the fact that each can be required individually by a TSL_2 grammar over an appropriate tier. In other words, TSL is not closed under intersection when the tier alphabets may differ. Interestingly, the set of strings containing exactly one instance of both 'a' and 'b' is recognized by a TSL_3 grammar since its projection to the $\{\mathrm{a}, \mathrm{b}\}$ tier is finite and thus SL:

$$\bigwedge\{\neg\rtimes\ltimes, \neg\rtimes\mathrm{a}\ltimes, \neg\rtimes\mathrm{b}\ltimes, \neg\mathrm{aa}, \neg\mathrm{bb}, \neg\mathrm{aba}, \neg\mathrm{bab}\}.$$

Although TSL is not in general closed under intersection, the following holds:

Theorem 2. *If $L_1 \in \mathrm{TSL}_{k_1}^\tau$ and $L_2 \in \mathrm{TSL}_{k_2}^\tau$, then the intersection $L_1 \cap L_2 \in \mathrm{TSL}_{\max(k_1,k_2)}^\tau$.*

Proof. Let L_1 and L_2 be as stated, and further let $L = L_1 \cap L_2$ and $k = \max(k_1, k_2)$. Then L is closed under insertion and deletion of symbols outside of τ because for any $w \in L$, by definition $w \in L_1$ and $w \in L_2$, and both of these sets are so closed. L is closed under substitution of preprojective suffixes by the same reasoning. Then by Theorem 1, L is TSL_k^τ. □

This theorem fails to hold for intersections of TSL stringsets over different tiers because the closure properties do not hold on both operands.

We can also show that TSL is not closed under union by demonstrating that the set of strings where all instances of 'a' precede all those of 'b' is TSL ($\neg$ba), and that where all instances of 'b' precede all those of 'a' is TSL ($\neg$ab), but their union is not:

$$\begin{array}{r} \rtimes\mathrm{b}\boxed{\mathrm{a}^{k-1}}\ltimes\phantom{\mathrm{b}} \\ \rtimes\boxed{\mathrm{a}^{k-1}}\mathrm{b}\ltimes \\ \hline \rtimes\mathrm{b}\boxed{\mathrm{a}^{k-1}}\mathrm{b}\ltimes. \end{array}$$

In general, to prove that a stringset is TSL one needs only provide the grammar. To show that a stringset cannot be TSL, one can use insertion or deletion closure to determine some symbols that must be on the tier alphabet and then use strings formed from only those symbols to demonstrate a failure of closure under substitution of preprojective suffixes. We leave as an exercise for the reader to show that TSL is not closed under complement, nor (since Σ^* is TSL) under relative complement.

6 Linguistic Examples

There are several TSL linguistic phenomena. Any Strictly k-Local pattern over an alphabet Σ can be described by a TSL_k^Σ grammar as well. Of course, this is uninteresting as we generally want to describe these phenomena with a lowest measure of complexity. The TSL class is motivated by the set of patterns that it can capture that SL does not.

One such pattern is the one-stress constraint described at the beginning of the previous section. The two sub-constraints that comprise it, namely that some syllable with primary stress occurs and that no more than one such syllable occurs, are coSL_1 (coSP_1) and $\mathrm{LTT}_{1,2}$ (SP_2), respectively, under standard adjacency and precedence accounts.

Though this constraint is neither purely SP nor purely SL, it can be described using TSL alone.

This particular kind of TSL constraint demonstrates applicability when long-distance dependencies are in effect. As another example, let us consider the simple SL_2 constraint that is alternation:

$$\bigwedge\{\neg\mathrm{ll}, \neg\mathrm{rr}\}.$$

If this constraint is applied on the tier of liquids (here only "l" and "r"), then the result is a dissimilation constraint like that of Latin as described by Cser (2010). The pattern described by Cser is a bit more involved, though: the dissimilation is blocked by non-coronal consonants. A TSL description accounts for these blockers: in addition to the liquids, the non-coronal consonants are on the tier as well.

Let us look now at an attested non-TSL pattern. In the previous section we described a method to prove that a given stringset is not TSL, and we will apply that here to the backness harmony in Uyghur as described by Mayer and Major (2018). The cited paper contains a full description of the pattern, but a simplification is as follows. Vowels and consonants both have harmonizing and transparent instances. Here we consider only the front/back vowel pair "y" and "u" and the back consonant "q". A suffix harmonizes to the rightmost harmonizing vowel if there is one, else with the rightmost harmonizing consonant if there is one, else the result is unspecified. We will ignore the final clause here.

We can prove that each of "y", "u", and "q" must be on the tier by constructing a stem of transparent segments and an affix that contains a harmonizing vowel. Inserting a harmonizing segment of mismatched backness into the stem causes an otherwise acceptable word to be rejected, and thus each of these three segment types must be on the tier. The following demonstrates a failure of closure under substitution of preprojective suffixes:

$$\begin{array}{c}
\rtimes\mathrm{y}\,\boxed{\mathrm{q}^k}\,\mathrm{y}\ltimes \\
\rtimes\mathrm{u}\,\boxed{\mathrm{q}^k}\,\mathrm{u}\ltimes \\
\hline
\rtimes\mathrm{y}\,\boxed{\mathrm{q}^k}\,\mathrm{u}\ltimes
\end{array}$$

It then follows that this pattern is not TSL for any choice of parameters.

In this section, we have again shown that the one-stress constraint and Latin liquid dissimilation are TSL by providing grammars with appropriate parameters, and we have used the methodology of the previous section to prove that backness harmony in Uyghur is not TSL for any parameters.

7 Multiple Relations, Additional Tiers

We proved earlier that TSL^{τ} is closed under intersection, but TSL in general is not. In this section, we discuss the intersection of TSL stringsets of incompatible relations (i.e. unequal tier alphabets). This is the complexity class inhabited by those stringsets that can be described by a coöccurrence of several TSL constraints operating over tier alphabets that are not necessarily equal.

The intersection of TSL^{τ_i} stringsets $(1 \leq i \leq n)$ is locally definable in the strict sense when each forbidden factor is considered with respect to its own relation. Operationally this would be equivalent to using n distinct projective tiers, a concept explored by De Santo and Graf (2019) and referred to as MTSL. For $T = \bigcup_{1 \leq i \leq n}(\tau_i)$, it is clear that insertion and deletion of symbols outside of T remains conservative. Yet T-preprojective suffix substitution no longer is; a slight modification is required in order to obtain this property:

Definition 2 (Generalized Preprojective Suffix Substitution)**.** For two strings $w_1 = u_1x_1v_1$ and $w_2 = u_2x_2v_2$ where:

$$\begin{aligned}
(\forall i)\big[\quad & (|x_1|^{\lhd[\tau_i]} \geq k-1 \\
& \quad \vee\ \pi_{\tau_i}(x_1) = \pi_{\tau_i}(w_1)) \\
& \wedge (|x_2|^{\lhd[\tau_i]} \geq k-1 \\
& \quad \vee\ \pi_{\tau_i}(x_2) = \pi_{\tau_i}(w_2)) \\
& \wedge\ \pi_{\tau_i}(x_1) = \pi_{\tau_i}(x_2) \qquad \big],
\end{aligned}$$

the string $w_3 = u_1x_1v_2$ is formed by the more general $\{\tau_i\}$*-preprojective suffix substitution.*

In words, on each tier, x_1 and x_2 have equal projections, which are either of length at least $k-1$ or equal to the projection of each word. This generalized operation is conservative, as the shared x substrings are guaranteed to have sufficiently many tier-symbols to allow for suffix-substitution on each projected tier. Therefore closure under this operation, and under insertion and deletion of symbols outside of the union of the tier alphabets, is necessary for a stringset to be in MTSL. Like the pumping lemma for Regular stringsets, lack of these closures can then be used to disprove class membership. It is provably not a characterization, which, like the Myhill-Nerode theorem, would allow the closures to constitute proof of membership.

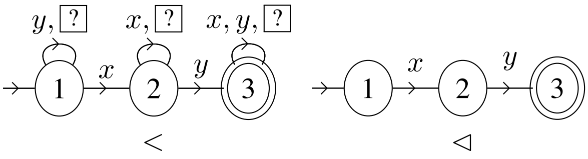


Figure 2: The factor $\rtimes xy \ltimes$ under multiple relations.

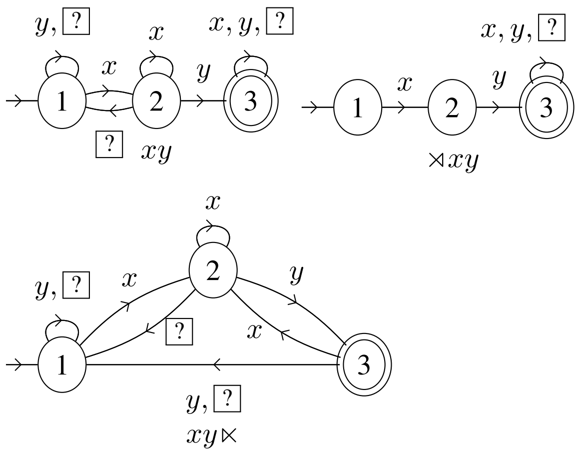


Figure 3: The factors xy, $\rtimes xy$, and $xy\ltimes$ under $\lhd$.

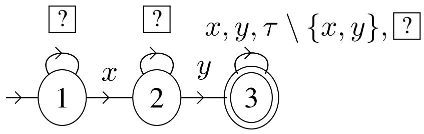


Figure 4: The factor $\rtimes xy$ under $\lhd^\tau$.

8 Logical Formulae and Automata

In this section, we briefly discuss the construction of finite-state automata for locally definable stringsets under each of the $\lhd$, $<$, and $\lhd^\tau$ relations (defining Local, Piecewise, and Tier-Local classes, respectively). In building automata that represent arbitrary logical formulae, one could either determine an appropriate alphabet beforehand or construct automata in such a way that only necessary symbols are considered. Here we use the latter approach, introducing a placeholder $\boxed{?}$ for potential other symbols. We define a DFA by a five-tuple $\mathcal{A} = \langle \Sigma, Q, \delta, q_\rtimes, F\rangle$ where Σ is an alphabet, Q a set of states, δ a (partial) transition function, $q_\rtimes$ an initial state, and F a set of final states.

The simplest case is the Piecewise formulae, as anchors do not affect $<$. For a string $\sigma_1 \ldots \sigma_n$ related by $<$, define $\Sigma = \{\sigma_1, \ldots, \sigma_n, \boxed{?}\}$ and construct a set of states $\{q_1, \ldots, q_{n+1}\}$ and a transition function of the form:

$$\delta(q_i, \sigma) = \begin{cases} q_{i+1} & \text{if } \sigma = \sigma_i \\ q_i & \text{otherwise.} \end{cases}$$

For $q_\rtimes = q_1$ and $F = \{q_{n+1}\}$, this reflects our intention, that the factor $\sigma_1 \ldots \sigma_n$ under $<$ occurs. Figure 2 shows the automaton constructed for the factor $\rtimes xy \ltimes$.

For factors defined using adjacency instead of precedence, we begin with fully anchored factors of the form $\rtimes \sigma_1 \ldots \sigma_n \ltimes$. The construction is the same as for Piecewise factors, except that the transition function is only defined for (q_i, σ_i). For factors that are not fully anchored, concatenate Σ^* to the side(s) missing an anchor (and determinize and minimize as appropriate). Figure 3 shows the less-anchored versions of $\rtimes xy \ltimes$.

In order to transform an adjacency factor into a tier-adjacency factor, note that the former is simply the projective image of the latter. Since the $\lhd^\tau$ relation does not attend to non-tier symbols, insertion of such a symbol at a given state must lead to a Nerode-equivalent state. Since the DFAs we are using here are minimal, it follows that each state should have a self-loop on all non-tier symbols. Thus we can first replace all instances of $\boxed{?}$ by parallel edges on each symbol in $\tau \setminus \{x, y\}$, and then add a self loop on $\boxed{?}$ to each state to account for symbols not on the tier. Figure 4 shows this transformation applied to the factor $\rtimes xy$.

Given these constructions for individual factors, unary operations such as the complement or iteration-closure are the standard automata-theoretic operations. For binary operations, given automata $\mathcal{A}_1$ and $\mathcal{A}_2$ whose alphabets are Σ_1 and Σ_2, add transitions on $\Sigma_2 \setminus \Sigma_1$ to $\mathcal{A}_1$ in parallel to all existing transitions on $\boxed{?}$ and similarly on $\Sigma_1 \setminus \Sigma_2$ to $\mathcal{A}_2$, then apply the standard automata-theoretic operation as usual. This use of a distinct placeholder symbol allows constraints to be defined by automata of minimal alphabet that expand in a way that preserves their semantics.

With these constructions, we can create DFAs for any stringsets definable by Boolean combinations of SL, SP, and TSL formulae, including among other things MTSL. Concatenation of automata for sequences of (Tier-)Local factors yields Piecewise-(Tier-)Local ones (Rogers and Lambert, 2019b). Boolean operations on these would yield Multi-Tier-Based Piecewise-Locally Testable stringsets: Boolean combinations of factors defined by occurrence, in order if not adjacently, of blocks of symbols on any of a number of projective tiers.

9 Deconstructing Automata

Since TSL^{τ} stringsets are closed under insertion of symbols not in τ, any transition on such a symbol from a given state must lead to a Nerode-equivalent state. Thus in a minimal DFA, such transitions are necessarily self-loops. Let $\mathcal{A} = \langle \Sigma, Q, \delta, q_{\rtimes}, F\rangle$ be a minimal DFA and define:

$$\bar{\tau} = \Big\{\sigma : (\forall q)\big[\delta(q,\sigma) = q\big]\Big\}.$$

The projection of $\mathcal{A}$ to τ ($\pi_{\tau}(\mathcal{A})$) is the result of replacing all transitions on symbols from $\bar{\tau}$ by transitions on ε, and since these transitions are all self-loops, this is equivalent to simply removing them. Then $\mathcal{A}$ represents a TSL_k^{τ} stringset iff this projection represents an SL_k one. The algorithms of Rogers and Lambert (2019a) can then be used to extract SL constraints from the projection, which of course are the TSL^{τ} constraints of $\mathcal{A}$ itself. Use of these algorithms provides a simple way to test whether an arbitrary Regular stringset is TSL_k^{τ}, and if so, for which parameters k and τ and even which grammar.

On the other hand, if $L(\mathcal{A})$ was not TSL, then since the extracted SL constraints describe the smallest SL superset of $L\big(\pi_{\tau}(\mathcal{A})\big)$, it follows that they then also describe the smallest TSL^{τ} superset of $L(\mathcal{A})$. That said, there may be smaller TSL supersets over different tiers.

10 Conclusions

The Tier-Based Strictly k-Local (TSL_k) class of stringsets was introduced by Heinz et al. (2011) and the question of what an abstract characterization for the class might be has remained open until now. We introduced here an abstract characterization, which can be used to provably state whether or not a given stringset is in the class. We then used this to prove various closure properties of the class itself. As TSL is not closed under intersection (but TSL^{τ} for fixed τ is), we discussed its intersection closure (MTSL) and provided a property that is necessary to be in MTSL. Failure to satisfy this property thus proves that a stringset is not in this class.

Further, to better integrate the TSL class with the other Piecewise-Local classes on the Subregular hierarchy, we introduced a tier-successor relation and associated logical formulae. We then described a method to construct deterministic finite-state automata from such formulae in order to harness the plentiful library of existing automata-theoretic tools. Finally, we used our abstract characterization to demonstrate a method of factoring a TSL automaton into individual constraints and a method of finding the constraints that produce the smallest TSL superset of a given non-TSL automaton. This provides a means to determine whether an arbitrary regular stringset is TSL_k^{τ}, and if so, for which parameters.

11 Future Work

We would like to explore linguistic applications of the Tier-Based extensions to the other classes in the piecewise-local subregular hierarchy, such as the Tier-Based Locally Testable stringsets hinted at in section 2 or the arbitrary formulae from section 8. For example, it would appear that Uyghur backness harmony might be MTLT, where the existence of harmonizing vowels can turn off the constraint referencing consonants.

Acknowledgments

The authors would like to thank Jeffrey Heinz at Stony Brook University for his contributions to the study of piecewise-local subregularity. We would also like to thank the anonymous reviewers for their detailed and useful feedback.

References

Pascal Caron. 1998. LANGAGE: A Maple package for automaton characterization of regular languages. In Derick Wood and Sheng Yu, editors, *Automata Implementation*, volume 1436 of *Lecture Notes in Computer Science*, pages 46–55. Springer Berlin / Heidelberg.

András Cser. 2010. The -alis/-aris allomorphy revisited. In Franz Rainer, Wolfgang Dressler, Dieter Kastovsky, and Hans Christian Luschützky, editors, *Variation and Change in Morphology: Selected Papers from the 13th International Morphology Meeting*, pages 33–52. John Benjamins Publishing Company, Vienna, Austria.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In Raffaella Bernardi, Greg Kobele, and Sylvain Pogodalla, editors, *Formal Grammar 2019*, volume 11668 of *Lecture Notes in Computer Science*, pages 35–50. Springer Verlag.

Pedro Garcia, Enrique Vidal, and José Oncina. 1990. Learning locally testable languages in the strict sense. In *Proceedings of the 1st International Workshop on Algorithmic Learning Theory*, pages 325–338, Tokyo, Japan.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, volume 2, pages 58–64, Portland, Oregon. Association for Computational Linguistics.

Larry M. Hyman. 2009. How (not) to do phonological typology: The case of pitch-accent. *Language Sciences*, 31(2–3):213–238.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In Annie Foret, Greg Kobele, and Sylvain Pogodalla, editors, *Formal Grammar 2018*, volume 10950 of *Lecture Notes in Computer Science*, pages 62–83.

Robert McNaughton and Seymour A. Papert. 1971. *Counter-Free Automata*. MIT Press.

James Rogers and Dakotah Lambert. 2019a. Extracting Subregular constraints from Regular stringsets. *Journal of Language Modelling*, 7(2):143–176.

James Rogers and Dakotah Lambert. 2019b. Some classes of sets of structures definable without quantifiers. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 63–77, Toronto, Canada. Association for Computational Linguistics.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3):329–342.

Imre Simon. 1975. Piecewise testable events. In Helmut Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer Verlag, Berlin.

Automating Gloss Generation in Interlinear Glossed Text

Angelina McMillan-Major
University of Washington / Seattle, USA
aymm@uw.edu

Abstract

Interlinear Glossed Text (IGT) is a rich data type produced by linguists for the purposes of presenting an analysis of a language's semantic and grammatical properties. I combine linguistic knowledge and statistical machine learning to develop a system for automatically annotating low-resource language data. I train a generative system for each language using on the order of 1000 IGT. The input to the system is the morphologically segmented source language phrase and its English translation. The system outputs the predicted linguistic annotation for each morpheme of the source phrase. The final system is tested on held-out IGT sets for Abui [abz], Chintang [ctn], and Matsigenka [mcb] and achieves 71.7%, 80.3%, and 84.9% accuracy, respectively.

1 Introduction

While language documentation has a long history, warnings from linguists such as Hale et al. (1992) and Krauss (1992) concerning language extinction have revitalized and expanded documentation efforts by communities and linguists, though there is still much work to be done (Seifart et al., 2018). According to Seifart et al. (2018), it can take 40 and 100 hours to transcribe an hour of recorded material, and even more time is required to analyze the language as a whole before annotating a single segment of the data collected. Given the decreasing language diversity in the world, there is an identified and immediate need for automated systems to assist in reducing the human hours spent on the documentation process.

While costly to produce, the glosses in IGT allow linguistic generalizations that are implicitly present in natural text to be explicitly available for natural language processing. In addition to supporting field linguists in collecting data, better and more easily produced IGT would also benefit end-stage projects such as machine translation between low-resource languages by improving the accuracy of the pre-processing modules (Xia and Lewis, 2008). Georgi et al. (2012) used IGT corpora to improve dependency parsing on low-resource languages using bootstrapping methods, while Bender et al. (2014) and Zamaraeva et al. (2019) used IGT to build high-precision grammars. Furthermore, language communities with trained IGT generators would be able to produce IGT for any new text found or created to aid with either language learning, documentation, or future translation efforts.

IGT consist of a source language phrase, a translation of that phrase into the language of the target audience, such as English, and glosses for each source morpheme. The glosses highlight the morphological and syntactic features of the source language. Ex. 1 shows an IGT from the Kazakh dataset in the Online Database of INterlinear text (ODIN) (Lewis and Xia, 2010), modified from Vinnitskaya et al. (2003).

(1) Kyz bolme-ge kir-di
girl.NOM room-DAT enter-PAST
(A/the) girl entered (a/the) room. [ISO 639-3: kaz]

In Ex. 1, the first line is the source line, the second is the gloss line, and the third is the translation line. The strings *girl*, *NOM*, *room*, etc. are all glosses, but glosses that refer to grammatical information, such as *NOM*, will be referred to as *grams* and the glosses that refer to semantically contentful information, such as *girl*, will be referred to as *stems*.

In this paper I describe a system for producing the gloss line of IGT automatically. I restrict my system to producing just the gloss line, given a morphologically segmented source line and its translation line. Morphological segmentation packages such as Morfessor (Creutz and Lagus, 2007) are widely available, and in the documentation setting translations may be provided by a native speaker consultant. This system could be used in combination with such resources. The input to the system at test time includes the morphemes in the segmented source line and the translation in the bottom line, and the target output is the gloss line.

This system does not, however, produce new analyses of the source language. Rather it is assumed that the linguistic analyses at all levels and the transliteration are already formalized by the documentary team. The system is then learning patterns from the analyses in the training data and reproducing the patterns when given new data. While the system can be trained on one set of analyses and tested on another, the performance will depend on the amount of variation between the analyses. This is especially significant in the low-resource setting, where each data instance contributes a relatively large amount of information as compared to each data instance in a high-resource setting.

A survey of the literature on IGT curation, augmentation and automation is provided in §2. In §3, I present the data used for developing and testing the system. §4 describes both the machine learning methods and the rule-based methods of this particular system, where the rule-based methods provide an implementation for handling out of vocabulary, also referred to as *OOV*, tokens. This section also includes an explanation of the evaluation metrics. §5 presents the results on the development and test languages, as well as a systematic error analysis. Finally, §6 discusses the challenges and limitations inherent in casting annotation as a classification task while exploring possible improvements to the current method for predicting OOV tokens.

2 Related Work

Approaches to IGT creation tools range in terms of how much input is required from the human annotator to yield the finished product. A widely used tool for documentation is FieldWorks Language Explorer (FLEx) (Baines, 2009). FLEx includes functionality for manually annotating interlinear text in addition to creating dictionaries and other language resources. The annotation software assists the user by retaining source-gloss pairs previously entered by the user and suggesting these glosses when the source morpheme appears again. The suggestions are not automatically constrained, however, so FLEx will suggest all previously seen glosses regardless of their likelihood given the local context unless the user explicitly provides the constraint information. By contrast the system presented here calculates the likelihood of a source morpheme being labeled with each possible gloss given the current sequence of morphemes and selects the most likely gloss automatically.

Palmer et al. (2009) (see also Baldridge and Palmer 2009 and Palmer et al. 2010) approached the task of IGT glossing within an active learning framework. In an active learning framework, annotators label the first small batch of input data, which is incorporated into the model in a new training phase, and then the next batch of data is labeled by the model and corrected by the annotators before being incorporated back into the model. They trained a maximum entropy classifier to predict a gloss given a morpheme and a context window of two morphemes before and after the morpheme in question. They had two annotators label IGT for Uspanteko [usp] (Mayan, Guatemala), using data from the OKMA corpus (Pixabaj et al., 2007). This corpus contains 32 glossed and 35 unglossed texts for a total of approximately 75,000 glossed tokens. They restrict the number of labels in the annotation schema by labeling stem morphemes with their part of speech (POS) tags, as provided in the corpus. Palmer et al. found that the expert annotator was more efficient and performed better when presented with the model's most uncertain predictions, but the naive annotator annotated more accurately when presented with random IGT rather than the most uncertain. These results suggest that active learning strategies must take the annotator into account in order to be optimally efficient, whereas automatic annotation does not have this constraint. Fully automated classification approaches provide an alternative method to IGT glossing when IGT have already been completed.

Samardžić et al. (2015) took a classification approach to IGT generation for the Chintang [ctn] (Kiranti, Nepal) Language Corpus dataset (Bickel et al., 2009). This corpus is significantly larger than the average documentation project with approximately 955,000 glossed tokens and a lexicon with POS tags. Samardžić et al. used two classifiers to generate their labels. The first classifier was based on Shen et al.'s (2007) version of

Collins and Roark's (2004) Perceptron learning algorithm and jointly learns the order in which to tag the sequence and the predicted tags. It annotated grammatical morphemes with their appropriate label and contentful morphemes with their POS tags, as in Palmer et al. (2009), to limit the total number of labels. The final step replaces the POS labels with an appropriate English lemma using the provided lexicon which maps English lemmas to Chintang morphemes. Samardžić et al. trained a trigram language model on the lexicon IDs to predict the most likely ID when multiple lemmas are possible, and back-off methods are used when labeling a previously unseen morpheme.

This paper attempts to add to the body of research on IGT generation by developing a machine learning framework that can apply to languages with fewer resources. Whereas these previous implementations rely on linguists' input or language specific resources, such as source language POS tags, to produce the final output, the system presented here runs using only what is given in the IGT training data. The following experiments attempt to answer the question of how much linguistic information statistical machine learning techniques are able to acquire from the linguistic patterns that are made explicit in IGT without any additional resources.

3 Data

The Online Database of INterlinear text (ODIN) is a repository of IGT examples collected from PDFs of linguistic publications (Lewis and Xia, 2010). ODIN contains 158,007 IGT from across 1,496 languages and 2,027 documents. The ODIN IGT datasets are stored in the XML-linearization of the Xigt format (Goodman et al., 2015), which includes a Python API.[1] A second version of ODIN[2] has been released with POS tags, dependency parses, and word alignments provided by the INterlinear Text ENrichment Toolkit (INTENT) system (Georgi, 2016).

I selected six languages from ODIN for developing the system based on set size: Turkish [tur], Russian [rus], Korean [kor], Japanese [jpn], Italian [ita], and Norwegian [nob]. I use a further three languages from language documentation projects as held-out test languages. Poor results on held-out languages compared to development languages would suggest that the system is inherently biased towards one language or one typological feature, such as word order; comparable results between the held-out and development languages provide evidence that the system performance is not dependent on language-specific features. The datasets for Chintang [ctn] (Kiranti, Nepal; Bickel et al. 2009), Abui [abz] (Trans-New Guinea, Indonesia; Kratochvíl 2017), and Matsigenka [mcb] (Maipurean, Peru; Michael et al. 2013) have been collected as part of language documentation projects and thus provide the opportunity to model system behavior in that setting. This setting typically includes consistent glossing schemes and native speaker consultants to provide translation information. In order for the system to produce models for these datasets in the same way as the ODIN datasets, preprocessing included converting the resources to the Xigt format and then enriching the data using the INTENT system (Georgi, 2016).

After filtering for IGT with identical source lines and IGT that were not fully annotated by INTENT, the Japanese and Korean sets have slightly more than 2000 IGT each, the Russian has set just under 1500 IGT, the Norwegian and Turkish sets have around 1000 IGT each, and the Italian set has around 800 IGT. Of the held-out datasets, Matsigenka is the smallest, with just under 450 IGT due to a large portion of the corpus having Spanish rather than English translations. The Abui and Chintang sets are much larger with approximately 4700 IGT and 7000 IGT.[3] For each language the system is trained using 90% of the given language's IGT and tested on the remaining 10%. Table 1 shows the number of IGT in each language's train and test sets from ODIN, while Table 2 shows the numbers for the held-out languages.

4 Methodology

I built one glossing system trained separately on each language dataset. Upon loading each dataset, the system removes IGT with source lines that appear multiple times in the dataset and IGT with missing or incomplete label references to the glosses and source morphemes. The system then

[1] http://github.com/xigt/xigt

[2] Available at http://depts.washington.edu/uwcl/odin/

[3] This is a subsample of the nearly 1 million word Chintang dataset (see §2).

formats the information in the remaining IGT to be fed into two Conditional Random Field (CRF) models (Lafferty et al., 2001). One model predicts the gloss line from the source line, hereafter referred to as the *source model* or *SRC model*, while the second model predicts the gloss line from the translation line, hereafter *translation model* or *TRS model*. Finally, the system incorporates the predictions of both models into the final output.

I use the Japanese example in (2), originally from Harley (1995), as a running example to show the steps in the system.

(2) yakko-ga wakko-o butai-ni agar-ase-ta
yakko-n wakko-a stage-on rise-cause-past
yakko made wakko rise onto the stage [jpn]

The source line, gold glosses, and the translation line are as they appear in the corpus.

4.1 Modeling

Conditional Random Fields (CRF) are able to classify sequences of tokens with a large number of possible labels while being sensitive to the context in which the tokens appear (Lafferty et al., 2001) and have been shown to be effective in low-resource settings (Ruokolainen et al., 2013). The CRF models were built using sklearn-crfsuite v0.3.6.[4] The training algorithm uses stochastic gradient descent with L2 regularization and a maximum of 50 iterations.

The SRC model predicts a gloss for each morpheme in the source line. When training, the system takes in complete IGT and uses the glosses provided as the gold training labels. The first whitespace-separated token in the source line is assumed to align with the first whitespace-separated token in the gloss line, the second source token with the second gloss, and so forth. While the SRC model is able to take advantage of the context provided by adjacent morphemes, it must also be provided with explicit features for source word boundaries. The features for each label include the source morpheme, the current source word, the previous and following words, and whether or not the previous and following morphemes are included in the current word (see Appendix A for an example). No processing of the source language, such as POS tags or dependency labels, other than the morphological segmentation has been assumed in this model, as many languages do not have access to NLP processing during the documentation process. At test time the SRC model would then output the following predicted sequence for the source line in Ex. 2:

(3) yakko-n pizza-acc taro-dat sit-cause-past

The second model, or TRS model, predicts the gloss that is aligned with each word in the translation line. The gold labels for the translation to gloss line predictions are provided by INTENT, which has automatically labeled the bilingual alignments between one gloss and one translation word. As a result, multi-word expressions are not considered in the TRS model unless they are explicit in the glosses. Many of the words in the translation line are not aligned with a gloss, so an additional null label is included. The features for each label include the translation word, its lemma as provided by the StanfordNLP API (Manning et al., 2014), and the POS tag and dependency structure for the translation word as provided by INTENT (again, see Appendix A for an example). The TRS model then outputs the following predicted sequence for the translation line in Ex. 2:

(4) yakko NA NA NA NA NA NA

NA stands for *Not Aligned* and is the most likely tag for the model to output. The content words that would be expected to be aligned in the translation line, *wakko*, *rise*, and *stage*, are not aligned in this case due to *wakko* and *rise* being OOV items, and *stage* having only been seen once in the training data. For further discussion of the TRS model's behavior, see §6. For both models, tokens that contain only punctuation are labeled with the gloss *PUNC*. Additionally, a dummy label is included in case of reference errors while accessing the data or when the features are not available. This may be the case with punctuation or with non-English words that the StanfordNLP lemmatizer is not able to process.

4.2 Integrating Model Hypotheses

At test time the given source line and its translation line are processed by their respective models. The output of each model is then assessed by the system. The system first checks whether the source tokens and their predicted glosses have co-occurred in the training data and whether the translation tokens and their predicted glosses have co-occurred in the training data. If a gloss is predicted

[4] http://github.com/TeamHG-Memex/sklearn-crfsuite

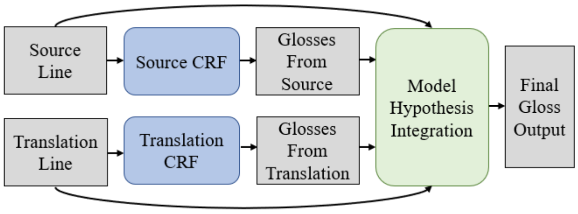


Figure 1: Visualization of the system

by both models and is supported by the training data, it's saved as a final prediction. If the SRC and TRS models disagree and the TRS model's prediction is supported by the training data, the TRS model's prediction is saved as the final prediction. If the original source token has been seen in the training data, but an exact match was not predicted by the translation line, the SRC model's prediction takes precedence. This is motivated by the fact that source tokens that are labeled with grams may not be aligned with a token in the translation.

If the source morpheme has not previously been seen, it is assumed to be a stem, and the lemma of an aligned translation lemma is used as the gloss (see § 6 for further discussion). If the source token is both unseen and unaligned, the system first checks to seen if there is an exact match between the morpheme and a translation word. Otherwise, the system separates the predicted grams, as identified by the gram list, from the SRC model's predicted gloss. Based on the grams, the system attempts to match the morpheme with a translation lemma with the same POS tag or argument role, using the grams to predict the morpheme's POS tag and the INTENT metadata to identify the translation words' POS tags or dependency structure. For example, if a case marker such as nominative is predicted, the system will look for a noun marked as the subject in the translation tokens. This process is implemented for nouns and verbs since OOV items are most likely to be in those categories. Finally, if the model is still unsure of the final prediction, the system selects the lemma of an unaligned translation word or the word itself if it cannot be lemmatized.

Continuing with the example from the previous section, the system now has the prediction information from Ex. 3 and 4. The system confirms that it has seen *yakko*, *ga*, *o*, *ni*, *ase*, and *ta* glossed as *yakko*, *n*, *acc*, *dat*, *cause*, and *past*, respectively, so it keeps those as final predictions. The system has seen *butai* in the training data but not glossed as *taro*, so it replaces the SRC model's prediction with the previously seen gloss, *stage*. The token *wakko* is an OOV item, but an exact match is found in the translation line, so the token itself is used as the gloss, replacing *pizza*. The token *agar* is also an OOV item, but because no grams were predicted by the TRS model, the system does not make any assumptions about the source POS tag and defaults to the token predicted by the SRC model. The resulting final prediction is:

(5) yakko-n wakko-acc stage-dat sit-cause-past

4.3 Evaluation

The system's performance is evaluated by comparing each gloss in each test IGT's final output to the gold standard glosses provided in the datasets. The system produces a label for each morpheme, so the recall provides no additional information. Comparing the final output in Ex. 5 with the gold gloss in Ex. 2, *yakko*, *n*, *wakko*, *stage*, *cause*, and *past* are correct for a total of 6/9. The system precision is given in terms of the micro-average over all tokens in all the IGT in each languages' test dataset.

I further analyze the system output by breaking down the system performance in terms of stems and grams. Labels are identified as grams or stems during the scoring process using a list of grams collected during the development of ODIN. The ODIN gram list covers many frequently used categories such as person, gender and case and has multiple realizations for each category's values.

There may be morpheme labels that contain multiple glosses, each separated by a period. In these cases, the predicted label is evaluated as a whole when scoring the system accuracy. When determining the system performance over stems and grams, however, the predicted label is split on each period and each gloss is checked against the ODIN gram list to determine if it is a gram or not. The gold label is also split if it contains at least one period. For each gloss in the gold label, if it is seen in the predicted label, it is considered correct, regardless of the order. Because the system may predict a label that has more or fewer glosses than the gold label, both the precision and recall are calculated. Each metric is presented in terms of the micro-average over all the stem tokens and the micro-average over all the gram tokens.

Ex. 2 does not contain any instances of a single label containing multiple glosses, so the combined

Lang. [ISO 639-3]	Train	Test	Acc
Japanese [jpn]	2062	229	77.8%
Korean [kor]	1956	217	75.6%
Norwegian [nob]	958	107	63.1%
Turkish [tur]	894	99	60.3%
Italian [ita]	732	81	59.9%
Russian [rus]	1322	147	53.2%

Table 1: Development languages, number of IGT training and test instances for each model, and test accuracy.

score for the stems and morphemes is not different from the morpheme score. In a more complicated example from the Japanese dataset originally from Bobaljik (n.d.), there are two instances of multi-gloss labels, *last.night* and *by.dat*.

(6) yuube kuruma-ga doroboo-ni nusum-are-ta
last.night car-nom robber-by.dat steal-pass-past
Last night, cars were stolen by a thief. [jpn]

The SRC model predicts the sequence *japanese car-nom thief-by steal-pass-past*. The TRS model predicts that *last*, *night* and *thief* are glosses. The rest of the words are not predicted to be aligned, and the final output is determined to be *last car-nom thief-by steal-pass-past*. In this output, the predicted label for *yuube* is missing a stem, *night*, *thief* is predicted instead of *robber*, and the predicted label for *ni* is missing a gram, *dat*. The morpheme score is 5/8, but the stem precision is 3/4, the gram precision is 4/4, the stem recall is 3/5, and the gram recall is 4/5.

5 Results

The results of all the development languages vary greatly, ranging from 53.2% to 77.8% accuracy.[5] There is a noticeable trend in which the relative model accuracy is predictable from the number of test IGT, with the exception of the Russian dataset. Table 1 shows the number of test IGT, training IGT, and system accuracy per development language. Table 2 shows the same information for the held-out languages, with an increasing number of training IGT over the same test set for Abui and Chintang. In addition to training on the full training sets, I also train the system on the initial 25%, 50%, and 75% of the training data for Abui and Chintang to see the effect of training set size on the system accuracy and train again on a random

[5] Code and instructions for reproducing these results are available at https://github.com/mcmillanmajora/IGTautoglossing.

Lang. [ISO 639-3]	Train	Test	Acc
Matsigenka [mcb]	388	43	**84.9%**
Chintang [ctn]	6589	677	**80.3%**
initial 75%	4941	677	**74.6%**
random 75%	4941	677	74.3%
initial 50%	3294	677	**72.6%**
random 50%	3294	677	72.5%
initial 25%	1646	677	68.7%
random 25%	1646	677	**69.0%**
Abui [abz]	4295	447	**71.7%**
initial 75%	3224	447	69.9%
random 75%	3224	447	**70.4%**
initial 50%	2149	447	68.7%
random 50%	2149	447	**69.1%**
initial 25%	1076	447	**66.1%**
random 25%	1076	447	64.9%

Table 2: Held-out languages, number of training and test IGT, and test accuracy. Training instances were selected randomly if *random* or from the beginning of the dataset if *initial*. Test IGT were held constant.

25%, 50%, and 75% of the training data to see the effect of vocabulary overlap. These datasets include IGT from different documentation sessions, so the assumption is that consecutive IGT are more likely to have been created at the same time and therefore contain repeated words. These sets are all tested using the same IGT in the test set for the full training data experiment.

5.1 Development Languages

Among the development languages, the system had the highest accuracies with the Korean and Japanese datasets at 75.6% and 77.8%. The Japanese training set had just over 2000 IGT and the Korean training set had just under 2000 IGT. Both sets had slightly more than 200 test IGT. The system performed less well over the Italian, Turkish and Norwegian datasets at 59.9%, 60.3%, and 63.1%, respectively. These datasets had less than half the data of the Japanese and Korean datasets. The system performed worst over the Russian dataset, at 53.2% accuracy on 1322 training instances, almost a third more than the Norwegian dataset.

A clearer pattern in the system's performance over the development languages arises when the labels are broken down into stems and grams, as seen in Table 3. For stems, precision scores range between 60.9% and 73.3% and recall scores range between 59.9% and 71.6%, whereas the precision

Lang.	Prec.		Rec.	
	Stem	Gram	Stem	Gram
jpn	73.3%	88.2%	71.6%	85.4%
kor	72.1%	83.0%	70.5%	80.5%
nob	63.8%	73.5%	62.7%	65.8%
tur	61.7%	63.8%	61.1%	56.1%
ita	63.6%	60.6%	62.6%	48.8%
rus	60.9%	67.4%	59.9%	49.2%

Table 3: Analysis of system performance on development languages with precision and recall for stems and grams.

Lang.	Prec.		Rec.	
	Stem	Gram	Stem	Gram
mcb	**73.5%**	**96.0%**	**70.3%**	**95.8%**
ctn	**71.2%**	**92.5%**	**69.9%**	**92.9%**
init. 75%	**60.7%**	**92.2%**	**60.4%**	**92.9%**
rand. 75%	60.5%	92.0%	59.9%	92.7%
init. 50%	57.2%	91.1%	56.2%	**92.2%**
rand. 50%	**57.3%**	91.1%	56.2%	92.1%
init. 25%	51.0%	88.7%	51.0%	91.1%
rand. 25%	**51.1%**	**89.0%**	**51.3%**	**91.3%**
abz	**70.3%**	**83.4%**	**72.5%**	**85.8%**
init. 75%	68.4%	81.9%	70.6%	84.5%
rand. 75%	**69.0%**	**82.7%**	**71.1%**	**85.1%**
init. 50%	66.9%	81.4%	68.8%	83.4%
rand. 50%	**67.8%**	**82.1%**	**69.5%**	**84.5%**
init. 25%	**63.4%**	79.6%	**65.6%**	**82.9%**
rand. 25%	63.0%	**79.9%**	65.0%	81.4%

Table 4: Analysis of system performance on held-out languages with precion and recall for stems and grams.

scores for grams range between 60.6% and 88.2% and the recall scores range between 48.8% and 85.4%. Japanese, Korean, and Norwegian all have higher scores for grams than stems in both precision and recall. That trend reverses for Turkish, Italian, and Russian, where the recall for grams is lower than stems. Japanese, Korean, and Turkish have much lower ratios of stems to grams, each having about 3 stem morphemes for every 2 grams. Russian, Italian, and Norwegian have about 5, 7, and 10 stems, respectively, for every 2 grams. Norwegian's high ratio is likely due to the syntactic similarity between it and English, which makes glossing with inflected English words easier. Because grams are often not annotated as separate morphemes, poor recall on grams would contribute to over lower scores on morpheme accuracy even if the stem is correctly predicted because the evaluation considers the predicted label as a whole. This is particularly true in the ODIN data, which also suffers from errors introduced when extracting IGT from linguistic papers and from what Lewis and Xia (2008) call *IGT bias*. IGT are most likely presented for a specific phenomena that is unique within the language and is overly represented in the paper compared to broader contexts. As a result, the set of IGT pulled from a single paper are likely skewed and the glossing may reflect the focus on a particular portion of the sentence, if a full sentence is given.

5.2 Held-out Languages

The system achieved higher accuracies over the Matsigenka and Chintang datasets than the development sets and comparable accuracies for the Abui dataset. The system achieved a higher accuracy for Matsigenka, 84.9%, than it did for any of the development datasets, which all had at least twice as much training data. The system was also trained for randomized and initial subsets of the training data for Abui and Chintang, resulting in 7 total experiments for each language. Table 2 shows the results on the various splits. The Abui results range from 66.1% to 71.7% on 447 test IGT, and the Chintang results range from 69% to 80.3% on 677 test IGT.

The held-out languages do pattern with the well-performing development datasets in terms of higher precision and recall for grams than stems. Table 4 shows the gram precision ranging from 79.6% to 96.0% and the gram recall ranging from 81.4% to 95.8% over all of the datasets. The stem scores have greater ranges, from 51% to 73.5% for precision and 51% to 72.5% for recall. The Chintang and Abui subsets do not differ more than 2% accuracy between the randomized and the non-randomized training set pairs. The Chintang stem precision and recall increase the most between the 75% and full sets, but the Abui stems see the biggest increase between the 25% and 50% subsets.

Samardžić et al. (2015) achieve 96% accuracy on 200,000 test word tokens in the Chintang dataset using approximately 800,000 word tokens for training. My system is maximally tested on 7250 Chintang morphemes using only 55,000 training morphemes and achieves 80.3% accuracy. My system also does not assume any language-specific metadata, while Samardžić et al. make use of a Chintang lexicon containing high-quality source POS tags. They also provide an analysis of their system's performance over lexical labels (stems) and functional labels (grams). In general, their model's performance over grams increases with the training set size, while the performance over stems remains fairly constant. Samardžić et al. attribute this pattern to the sequential inclusion of IGT collected from source texts that differ lexically or stylistically as well as differing annotation schema over these sources.

6 Error Analysis

In investigating the predictions made by the models and the final output glosses, a number of inconsistencies in the ODIN datasets became apparent. Processing errors occur when there are a mismatched number of source morphemes and gloss labels, such as when a multi-word expression is used as a single gloss and contains whitespace or when a coindexation variable is included in the source line as a separate token. Some instances also include additional punctuation indicating clausal boundaries. Authors of linguistic papers use IGT to illustrate syntactic and semantic properties of languages and these additional annotations are often included to highlight the relevant information for the audience.

Due to the wide range of authors from which the ODIN IGT originate, many grams may refer to the same grammatical concept, as shown in Ex. 2 and 6 from the Japanese dataset. The morpheme *ga* indicates the nominative case, but is labeled as *n* in Ex. 2 and *nom* in Ex. 6. The system treats these labels as unique though they are intended to be synonymous. In contrast to the unintended ambiguity, Ex. 7 and Ex. 9 below both contain the Chintang morpheme *lo*, but in Ex. 6 it is labeled as *okay* and in Ex. 8 it is labeled *surp* as in the morpheme indicates the speaker's surprise.

(7) lo sat na maha na
okay seven top not top
okay, not seven [ctn] (Bickel et al., 2009)

Furthermore, *lo* can also appear as a nominative suffix for the interrogative pronoun *sa*, meaning *who* (Paudyal, 2015). While these functions are difficult for the system to differentiate, it can learn the contexts for each function given enough examples and consistent annotation. Multiple labels for the same function, however, will cause the system to try to discriminate between instances of the same context, as in the case of the truly ambiguous morphemes. Furthermore, the high accuracy over the test languages suggests that the consistency of the annotations has a stronger effect on the system performance than dataset size.

The system also contributes a number of consistent errors. For example, in this IGT from the Korean dataset the system relies too heavily on the source line, ignoring the correct TRS model predictions.

(8) emeni-ka us-usi-ess-up-nita
mother-nom smile-sh-pst-pol-dec
mother smiled [kor] (Yang, 1994)

The SRC model predicts the sequence *mother-nom miss-hon-pst-pol-dec* and the TRS model predicts that *mother* and *smile* are glosses, however the system keeps the incorrect gloss *miss* from the SRC model because *us* and *miss* co-occurred in the training data. This suggests that overall system performance might improve if the source predictions were preferred for grams and the translation predictions for stems.

However, across all of the languages, the TRS model frequently predicts only the null label, as seen in Ex. 4. The training data alignments sometimes do not include alignments between grams and English function words, so a significant portion of the information in the translation line is not incorporated into the model. Including a preprocessing step to supplement the INTENT alignments by aligning English function words with likely glosses, such as *was* and *past*, may improve the TRS model accuracy by decreasing the likelihood of the null label.

Further improvements could also be made in the selection and lemmatization of OOV replacements from the translation. The system often fails to find the correct stem, and even when it does find the stem, it may not be a direct match with gold gloss.

(9) yo-ni terso lo nang
dem.across-dir straight surp but
there straightly [ctn] (Bickel et al., 2009)

In predicting the glosses for the source line in Ex. 9, the SRC model outputs the sequence *dem.across-dir really surp but*. The system identifies *terso* and *straightly* as OOV items, but fails to lemmatize *straightly* to *straight*.

This example also shows that the stem and gram scores for the held-out languages are not entirely accurate, as the non-ODIN annotations contain grams like *surp* not covered by the ODIN gram list. While this doesn't affect the the overall morpheme score, it may indicate that the patterns seen in the held-out data stem and gram scores don't reflect the system's true performance as reliably as the patterns over the development data. Allowing for project-specific gram lists may improve and provide more confidence in gram and stem scores.

The differing annotation schemata also make it difficult to draw cross-linguistic conclusions as each annotation schema is founded in a different set of theoretical assumptions. These experiments, however, do show some of the challenges that machine learning techniques have with language as a data type as opposed to other sequential data. Because of the learning algorithm's reliance on the surrounding context of each label to make predictions, the linguistic properties that introduce more possible answers to a morpheme's label due to ambiguous contexts make the predictions more difficult. For example, non-concatenative morphology, highly polysemous source morphemes, and irregularities in word order will all compound to make the information that the algorithm is able to learn from the training data more sparse. All languages contain these complexities to some degree, but the amount that is present in the training data will have a large effect on the system performance.

7 Future Work

Over all the languages, the system performance would improve by modifying how the system balances the information from the SRC and TRS models. Providing confidence scores for each predicted gloss and reducing the influence of the SRC model are immediate steps toward better accuracies. A pretrained TRS model over multiple language datasets may also minimize the number of OOV items in the model, thereby increasing the confidence of non-null glosses. Georgi (2016) saw a boost in the precision of alignments between the gloss line and the translation line using this technique with a statistical aligner, though the heuristic approach ultimately had a better F1 score due to higher recall. Georgi proposed that this was due to the variable word order of the gloss line when combining data from across languages, which suggests that the classification approach may be more robust to this variation as the model is learning the mapping from the translation word to the gloss rather than the alignment itself.

While the current implementation focuses on English translations, the submodules for POS tagging and dependency parsing could be modified to support documentation efforts using other high-resource languages. Further modification of the feature input system would allow users to make use of any additional resources available to their project. Confidence scores on all output labels would also help the end user in quickly identifying possible OOV or ambiguous tokens.[6] Once the model performance has been optimized over the available datasets, the true test of the system would be to monitor usability and its effect on the number of human hours required in an ongoing documentation project, as in Palmer et al. (2009).

8 Conclusion

This work outlines an initial supervised system for automatically annotating IGT given a morpheme-segmented source phrase and its translation. The system uses CRFs to predict the glosses from the source and translation lines individually and combines the information in a heuristic fashion to form a final prediction. The system was developed on six languages from ODIN, and tested on held-out languages. The held-out language datasets were provided by linguists and native speaker collaborators, modeling the intended use case of a documentation project. An intrinsic evaluation shows that system performs better on the held-out language datasets than the development data from ODIN, but the error analysis suggests that this is due to differences in annotation practices. Further work is needed to improve the system's final prediction selection, particularly with regards to OOV items.

Acknowledgments

Thank you to Emily Bender, Fei Xia, Michael Goodman, Ryan Georgi, David Inman, Olga Zamaraeva, Kristen Howell, and the anonymous reviewers for their comments and contributions to this work.

References

David Baines. 2009. Fieldworks Language Explorer (FLEx). *eLEX2009*, page 27.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning *actually* work? Time-based

[6]Thank you to an anonymous reviewer for this suggestion.

evaluation of cost-reduction strategies for language documentation. In *Proceedings of EMNLP 2009*.

Emily M Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 2014. Learning grammar specifications from IGT: A case study of Chintang. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 43–53.

Balthasar Bickel, Goma Banjade, Toya N Bhatta, Martin Gaenzle, Netra P Paudyal, Manoj Rai, Novel Kishore Rai, Ichchha Purna Rai, and Sabine Stoll. 2009. Audiovisual corpus of the Chintang language, including a longitudinal corpus of language acquisition by six children, plus a trilingual dictionary, paradigm sets, grammar sketches, ethnographic descriptions, and photographs. DoBeS, Universität Leipzig, Nijmegen, Leipzig.

Jonathan David Bobaljik. n.d. 321 Syntax I Lecture Notes: Class 4: NP-Movement.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34.

Ryan Georgi. 2016. *From Aari to Zulu : massively multilingual creation of language tools using interlinear glossed text*. Ph.D. thesis, University of Washington, Seattle, WA, USA.

Ryan Georgi, Fei Xia, and William Lewis. 2012. Improving dependency parsing with interlinear glossed text and syntactic projection. In *Proceedings of COLING 2012: Posters*, pages 371–380. The COLING 2012 Organizing Committee.

Michael Wayne Goodman, Joshua Crowgey, Fei Xia, and Emily M. Bender. 2015. Xigt: extensible interlinear glossed text for natural language processing. *Language Resources and Evaluation*, 49(2):455–485.

Ken Hale, Michael Krauss, Lucille J. Watahomigie, Akira Y. Yamamoto, Colette Craig, LaVerne Masayesva Jeanne, and Nora C. England. 1992. Endangered languages. *Language*, 68(1):1–42.

Heidi Britton Harley. 1995. *Subjects, events, and licensing*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Frantisek Kratochvíl. 2017. Abui corpus. electronic database: 162,000 words of natural speech, and 37,500 words of elicited material.

Michael Krauss. 1992. The worlds languages in crisis. *Language*, 68(1):4–10.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

William D Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, pages 685–690, Hyderabad, India.

William D. Lewis and Fei Xia. 2010. Developing ODIN: A multilingual repository of annotated language data for hundreds of the world's languages. *Literary and Linguistic Computing*, 25(3):303–319.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Lev Michael, Christine Beier, Zachary O'Hagan, Harold Vargas Pereira, and Jose Vargas Pereira. 2013. Matsigenka text written by Matsigenka authors.

Alexis Palmer, Taesun Moon, and Jason Baldridge. 2009. Evaluating automation strategies in language documentation. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 36–44, Boulder, Colorado. Association for Computational Linguistics.

Alexis Palmer, Taesun Moon, Jason Baldridge, Katrin Erk, Eric Campbell, and Telma Can. 2010. Computational strategies for reducing annotation effort in language documentation. *Linguistic Issues in Language Technology*, 3(4):1–42.

Netra P Paudyal. 2015. *Aspects of Chintang syntax*. Ph.D. thesis, University of Zurich, Zurich, Switzerland.

Telma Can Pixabaj, Miguel Angel Vicente Méndez, Marıa Vicente Méndez, and Oswaldo Ajcot Damián. 2007. Text collections in Four Mayan Languages. *Archived in The Archive of the Indigenous Languages of Latin America*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, Sofia, Bulgaria. Association for Computational Linguistics.

Tanja Samardžić, Robert Schikowski, and Sabine Stoll. 2015. Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 68–72.

Frank Seifart, Nicholas Evans, Harald Hammarstrm, and Stephen C. Levinson. 2018. Language documentation twenty-five years on. *Language*, 94(4):E324–E345.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague, Czech Republic. Association for Computational Linguistics.

Inna Vinnitskaya, Suzanne Flynn, and Claire Foley. 2003. The acquisition of relative clauses in a third language: comparing adults and children. In *Proceedings of the 6th Generative Approaches to Second Language Acquisition Conference*, pages 340–345.

Fei Xia and William D. Lewis. 2008. Repurposing theoretical linguistic data for tool development and search. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Byong-Seon Yang. 1994. *Morphosyntactic phenomena of Korean in role and reference grammar: psych-verb constructions, inflectional verb morphemes, complex sentences, and relative clauses*. Ph.D. thesis, State University of New York at Buffalo, Buffalo, NY, USA.

Olga Zamaraeva, Kristen Howell, and Emily M Bender. 2019. Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, page 5.

A Model Features

Using Ex. 2 as an illustration of the tagging process at test time, the system takes the source line as input then formats it to be fed into the SRC model. The representations for the first three morphemes can be seen in Table 5, where i is the current position in the sequence, m_i is the current morpheme, w_i is the current word, w_{i-1} is the previous word, m_{i+1} in w_i is the following morpheme if it occurs within the same word as m_i, and so on. The value *BOS* refers to the beginning of the sentence, and the value for the w_{i+1} feature for phrase-final morphemes is *EOS*, which refers to the end of the sentence.

feat. name	$i = m_1$	$i = m_2$	$i = m_3$...
m_i	yakko	ga	wakko
w_i	yakko-ga	yakko-ga	wakko-o
w_{i-1}	BOS	BOS	yakko-ga
w_{i+1}	wakko-o	wakko-o	butai-ni
m_{i-1} in w_i	NONE	yakko	NONE
m_{i+1} in w_i	ga	NONE	o

Table 5: Feature representation of the source line.

Again using Ex. 2, the TRS model would take the translation line as input and format it to be fed into the model. The representations for the first three words can be seen in Table 6, where i is the current position in the sequence, tw_i is the current translation word, ds_i is the dependency structure tag of the current word as given by the INTENT system, ps_i is the POS tag as given by INTENT, and lem_i is the lemma of the word as given by the StanfordNLP lemmatizer.

feat. name	$i = tw_1$	$i = tw_2$	$i = tw_3$...
tw_i	yakko	made	wakko
ds_i	nsubj	root	dobj
ps_i	nnp	vbd	nnp
lem_i	yakko	make	wakko

Table 6: Feature representation of the translation line.

What do you mean, BERT?
Assessing BERT as a Distributional Semantics Model

Timothee Mickus
Université de Lorraine
CNRS, ATILF
tmickus@atilf.fr

Denis Paperno
Utrecht University
d.paperno@uu.nl

Mathieu Constant
Université de Lorraine
CNRS, ATILF
mconstant@atilf.fr

Kees van Deemter
Utrecht University
c.j.vandeemter@uu.nl

Abstract

Contextualized word embeddings, i.e. vector representations for words in context, are naturally seen as an extension of previous noncontextual distributional semantic models. In this work, we focus on BERT, a deep neural network that produces contextualized embeddings and has set the state-of-the-art in several semantic tasks, and study the semantic coherence of its embedding space. While showing a tendency towards coherence, BERT does not fully live up to the natural expectations for a semantic vector space. In particular, we find that the position of the sentence in which a word occurs, while having no meaning correlates, leaves a noticeable trace on the word embeddings and disturbs similarity relationships.

1 Introduction

A recent success story of NLP, BERT (Devlin et al., 2018) stands at the crossroad of two key innovations that have brought about significant improvements over previous state-of-the-art results. On the one hand, BERT models are an instance of contextual embeddings (McCann et al., 2017; Peters et al., 2018), which have been shown to be subtle and accurate representations of words within sentences. On the other hand, BERT is a variant of the Transformer architecture (Vaswani et al., 2017) which has set a new state-of-the-art on a wide variety of tasks ranging from machine translation (Ott et al., 2018) to language modeling (Dai et al., 2019). BERT-based models have significantly increased state-of-the-art over the GLUE benchmark for natural language understanding (Wang et al., 2019b) and most of the best scoring models for this benchmark include or elaborate on BERT. Using BERT representations has become in many cases a new standard approach: for instance, all submissions at the recent shared task on gendered pronoun resolution (Webster et al., 2019) were based on BERT. Furthermore, BERT serves both as a strong baseline and as a basis for a fine-tuned state-of-the-art word sense disambiguation pipeline (Wang et al., 2019a).

Analyses aiming to understand the mechanical behavior of Transformers in general, and BERT in particular, have suggested that they compute word representations through implicitly learned syntactic operations (Raganato and Tiedemann, 2018; Clark et al., 2019; Coenen et al., 2019; Jawahar et al., 2019, a.o.): representations computed through the 'attention' mechanisms of Transformers can arguably be seen as weighted sums of intermediary representations from the previous layer, with many attention heads assigning higher weights to syntactically related tokens (however, contrast with Brunner et al., 2019; Serrano and Smith, 2019).

Complementing these previous studies, in this paper we adopt a more theory-driven lexical semantic perspective. While a clear parallel was established between 'traditional' noncontextual embeddings and the theory of distributional semantics (a.o. Lenci, 2018; Boleda, 2019), this link is not automatically extended to contextual embeddings: some authors (Westera and Boleda, 2019) even explicitly consider only "context-invariant" representations as distributional semantics. Hence we study to what extent BERT, as a contextual embedding architecture, satisfies the properties expected from a natural contextualized extension of distributional semantics models (DSMs).

DSMs assume that meaning is derived from use in context. DSMs are nowadays systematically represented using vector spaces (Lenci, 2018). They generally map each word in the domain of the model to a numeric vector on the basis of distributional criteria; vector components are inferred from text data. DSMs have also been computed for linguistic items other than words, e.g., word senses—based both on meaning inventories (Rothe and Schütze, 2015) and word sense induction techniques (Bartunov et al., 2015)—or meaning exemplars (Reisinger and Mooney, 2010; Erk and Padó, 2010; Reddy et al., 2011). The default approach has however been to produce representations for word types. Word properties encoded by DSMs vary from morphological information (Marelli and Baroni, 2015; Bonami and Paperno, 2018) to geographic information (Louwerse and Zwaan, 2009), to social stereotypes (Bolukbasi et al., 2016) and to referential properties (Herbelot and Vecchi, 2015).

A reason why contextualized embeddings have not been equated to distributional semantics may lie in that they are "functions of the entire input sentence" (Peters et al., 2018). Whereas traditional DSMs match word *types* with numeric vectors, contextualized embeddings produce distinct vectors per *token*. Ideally, the contextualized nature of these embeddings should reflect the semantic nuances that context induces in the meaning of a word—with varying degrees of subtlety, ranging from broad word-sense disambiguation (e.g. 'bank' as a river embankment or as a financial institution) to narrower subtypes of word usage ('bank' as a corporation or as a physical building) and to more context-specific nuances.

Regardless of how apt contextual embeddings such as BERT are at capturing increasingly finer semantic distinctions, we expect the contextual variation to preserve the basic DSM properties. Namely, we expect that the space structure encodes meaning similarity and that variation within the embedding space is semantic in nature. Similar words should be represented with similar vectors, and only semantically pertinent distinctions should affect these representations. We connect our study with previous work in section 2 before detailing the two approaches we followed. First, we verify in section 3 that BERT embeddings form natural clusters when grouped by word types, which on any account should be groups of similar words and thus be assigned similar vectors. Second, we test in sections 4 and 5 that contextualized word vectors do not encode semantically irrelevant features: in particular, leveraging some knowledge from the architectural design of BERT, we address whether there is no systematic difference between BERT representations in odd and even sentences of running text—a property we refer to as *cross-sentence coherence*. In section 4, we test whether we can observe cross-sentence coherence for single tokens, whereas in section 5 we study to what extent incoherence of representations across sentences affects the similarity structure of the semantic space. We summarize our findings in section 6.

2 Theoretical background & connections

Word embeddings have been said to be 'all-purpose' representations, capable of unifying the otherwise heterogeneous domain that is NLP (Turney and Pantel, 2010). To some extent this claim spearheaded the evolution of NLP: focus recently shifted from task-specific architectures with limited applicability to universal architectures requiring little to no adaptation (Radford, 2018; Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019; Liu et al., 2019, a.o.).

Word embeddings are linked to the distributional hypothesis, according to which "you shall know a word from the company it keeps" (Firth, 1957). Accordingly, the meaning of a word can be inferred from the effects it has on its context (Harris, 1954); as this framework equates the meaning of a word to the set of its possible usage contexts, it corresponds more to holistic theories of meaning (Quine, 1960, a.o.) than to truth-value accounts (Frege, 1892, a.o.). In early works on word embeddings (Bengio et al., 2003, e.g.), a straightforward parallel between word embeddings and distributional semantics could be made: the former are distributed representations of word meaning, the latter a theory stating that a word's meaning is drawn from its distribution. In short, word embeddings could be understood as a vector-based implementation of the distributional hypothesis. This parallel is much less obvious for contextual embeddings: are constantly changing representations truly an apt description of the meaning of a word?

More precisely, the literature on distributional semantics has put forth and discussed many mathematical properties of embeddings: embeddings are equivalent to count-based matrices (Levy and Goldberg, 2014b), expected to be linearly dependant (Arora et al., 2016), expressible as a unitary matrix (Smith et al., 2017) or as a perturbation of an identity matrix (Yin and Shen, 2018). All these properties have however been formalized for non-contextual embeddings: they were formulated using the tools of matrix algebra, under the assumption that embedding matrix rows correspond to word types. Hence they cannot be applied as such to contextual embeddings. This disconnect in the literature leaves unanswered the question of what consequences there are to framing contextualized embeddings as DSMs.

The analyses that contextual embeddings have been subjected to differ from most analyses of distributional semantics models. Peters et al. (2018) analyzed through an extensive ablation study of ELMo what information is captured by each layer of their architecture. Devlin et al. (2018) discussed what part of their architecture is critical to the performances of BERT, comparing pretraining objectives, number of layers and training duration. Other works (Raganato and Tiedemann, 2018; Hewitt and Manning, 2019; Clark et al., 2019; Voita et al., 2019; Michel et al., 2019) have introduced specific procedures to understand how attention-based architectures function on a mechanical level. Recent research has however questioned the pertinence of these attention-based analyses (Serrano and Smith, 2019; Brunner et al., 2019); moreover these works have focused more on the inner workings of the networks than on their adequacy with theories of meaning.

One trait of DSMs that is very often encountered, discussed and exploited in the literature is the fact that the relative positions of embeddings are not random. Early vector space models, by design, required that word with similar meanings lie near one another (Salton et al., 1975); as a consequence, regions of the vectors space describe coherent semantic fields.[1] Despite the importance of this characteristic, the question whether BERT contextual embeddings depict a coherent semantic space on their own has been left mostly untouched by papers focusing on analyzing BERT or Transformers (with some exceptions, e.g. Coenen et al., 2019). Moreover, many analyses of how meaning is represented in attention-based networks or contextual embeddings include "probes" (learned models such as classifiers) as part of their evaluation setup to 'extract' information from the embeddings (Peters et al., 2018; Tang et al., 2018; Coenen et al., 2019; Chang and Chen, 2019, e.g.). Yet this methodology has been criticized as potentially conflicting with the intended purpose of studying the representations themselves (Wieting and Kiela, 2019; Cover, 1965); cf. also Hewitt and Liang (2019) for a discussion. We refrain from using learned probes in favor of a more direct assessment of the coherence of the semantic space.

3 Experiment 1: Word Type Cohesion

The trait of distributional spaces that we focus on in this study is that similar words should lie in similar regions of the semantic space. This should hold all the more so for identical words, which ought to be be maximally similar. By design, contextualized embeddings like BERT exhibit variation within vectors corresponding to identical word types. Thus, if BERT is a DSM, we expect that word types form natural, distinctive clusters in the embedding space. Here, we assess the coherence of word type clusters by means of their *silhouette scores* (Rousseeuw, 1987).

3.1 Data & Experimental setup

Throughout our experiments, we used the Gutenberg corpus as provided by the NLTK platform, out of which we removed older texts (King John's Bible and Shakespeare). Sentences are enumerated two by two; each pair of sentences is then used as a distinct input source for BERT. As we treat the BERT algorithm as a black box, we retrieve only the embeddings from the last layer, discarding all intermediary representations and attention weights. We used the `bert-large-uncased` model in all experiments[2]; therefore most of our experiments are done on word-pieces.

To study the basic coherence of BERT's semantic space, we can consider types as clusters of tokens—i.e. specific instances of contextualized embeddings—and thus leverage the tools of cluster analysis. In particular, silhouette score is generally used to assess whether a specific observation $\vec{v}$ is well assigned to a given cluster C_i drawn from a set of possible clusters C. The silhouette score is defined in eq. 1:

$$\begin{aligned} \mathit{sep}(\vec{v}, C_i) &= \min\{\underset{\vec{v}' \in C_j}{\text{mean}}\, d(\vec{v}, \vec{v}')\forall\, C_j \in C - \{C_i\}\} \\ \mathit{coh}(\vec{v}, C_i) &= \underset{\vec{v}' \in C_i - \{\vec{v}\}}{\text{mean}}\, d(\vec{v}, \vec{v}') \\ \mathit{silh}(\vec{v}, C_i) &= \frac{\mathit{sep}(\vec{v}, C_i) - \mathit{coh}(\vec{v}, C_i)}{\max\{\mathit{sep}(\vec{v}, C_i),\ \mathit{coh}(\vec{v}, C_i)\}} \end{aligned} \tag{1}$$

We used Euclidean distance for d. In our case, observations $\vec{v}$ therefore correspond to tokens (that is, *word-piece* tokens), and clusters C_i to types.

[1] Vectors encoding contrasts between words are also expected to be coherent (Mikolov et al., 2013b), although this assumption has been subjected to criticism (Linzen, 2016).

[2] Measurements were conducted before the release of the `bert-large-uncased-whole-words` model.

Silhouette scores consist in computing for each vector observation $\vec{v}$ a cohesion score (viz. the average distance to other observations in the cluster C_i) and a separation score (viz. the minimal average distance to other observations, i.e. the minimal 'cost' of assigning $\vec{v}$ to any other cluster than C_i). Optimally, cohesion is to be minimized and separation is to be maximized, and this is reflected in the silhouette score itself: scores are defined between -1 and 1; -1 denotes that the observation $\vec{v}$ should be assigned to another cluster than C_i, whereas 1 denotes that the observation $\vec{v}$ is entirely consistent with the cluster C_i. Keeping track of silhouette scores for a large number of vectors quickly becomes intractable, hence we use a slightly modified version of the above definition, and compute separation and cohesion using the distance to the average vector for a cluster rather than the average distance to other vectors in a cluster, as suggested by Vendramin et al. (2013). Though results are not entirely equivalent as they ignore the inner structure of clusters, they still present a gross view of the consistency of the vector space under study.

We do note two caveats with our proposed methodology. Firstly, BERT uses subword representations, and thus BERT tokens do not necessarily correspond to words. However we may conjecture that some subwords exhibit coherent meanings, based on whether they tightly correspond to morphemes—e.g. '*##s*', '*##ing*' or '*##ness*'. Secondly, we group word types based on character strings; yet only monosemous words should describe perfectly coherent clusters—whereas we expect some degree of variation for polysemous words and homonyms according to how widely their meanings may vary.

3.2 Results & Discussion

We compared cohesion to separation scores using a paired Student's t-test, and found a significant effect (p-value $< 2 \cdot 2^{-16}$). This highlights that cohesion scores are lower than separation scores. The effect size as measured by Cohen's d (Cohen's $d = -0.121$) is however rather small, suggesting that cohesion scores are only 12% lower than separation scores. More problematically, we can see in figure 1 that 25.9% of the tokens have a negative silhouette score: one out of four tokens would be better assigned to some other type than the one they belong to. When aggregating scores by types,

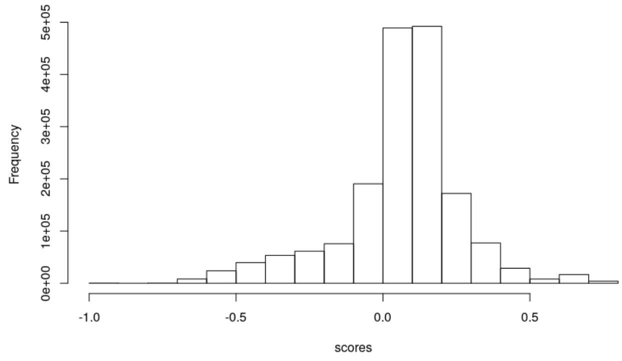


Figure 1: Distribution of token silhouette scores

we found that 10% of types contained only tokens with negative silhouette score.

The standards we expect of DSMs are not always upheld strictly; the median and mean score are respectively at 0.08 and 0.06, indicating a general trend of low scores, even when they are positive. We previously noted that both the use of subword representations in BERT as well as polysemy and homonymy might impact these results. The amount of meaning variation induced by polysemy and homonymy can be estimated by using a dictionary as a sense inventory. The number of distinct entries for a type serves as a proxy measure of how much its meaning varies in use. We thus used a linear model to predict silhouette scores with log-scaled frequency and log-scaled definition counts, as listed in the Wiktionary, as predictors. We selected tokens for which we found at least one entry in the Wiktionary, out of which we then randomly sampled 10000 observations. Both definition counts and frequency were found to be significant predictors, leading the silhouette score to decrease. This suggests that polysemy degrades the cohesion score of the type cluster, which is compatible with what one would expect from a DSM. We moreover observed that monosemous words yielded higher silhouette scores than polysemous words ($p < 2 \cdot 2^{-16}$, Cohen's $d = 0.236$), though they still include a substantial number of tokens with negative silhouette scores.

Similarity also includes related words, and not only tokens of the same type. Other studies (Vial et al., 2019; Coenen et al., 2019, e.g.) already stressed that BERT embeddings perform well on word-level semantic tasks. To directly assess whether BERT captures this broader notion of similarity, we used the MEN word similarity dataset

(Bruni et al., 2014), which lists pairs of English words with human annotated similarity ratings. We removed pairs containing words for which we had no representation, leaving us with 2290 pairs. We then computed the Spearman correlation between similarity ratings and the cosine of the average BERT embeddings of the two paired word types, and found a correlation of 0.705, showing that cosine similarity of average BERT embeddings encodes semantic similarity. For comparison, a word2vec DSM (Mikolov et al., 2013a, henceforth W2V) trained on BooksCorpus (Zhu et al., 2015) using the same tokenization as BERT achieved a correlation of 0.669.

4 Experiment 2: Cross-Sentence Coherence

As observed in the previous section, overall the word type coherence in BERT tends to match our basic expectations. In this section, we do further tests, leveraging our knowledge of the design of BERT. We detail the effects of jointly using *segment encodings* to distinguish between paired input sentences and *residual connections*.

4.1 Formal approach

We begin by examining the architectural design of BERT. We give some elements relevant to our study here and refer the reader to the original papers by Vaswani et al. (2017) and Devlin et al. (2018), introducing Transformers and BERT, for a more complete description. On a formal level, BERT is a deep neural network composed of superposed layers of computations. Each layer is composed of two "sub-layers": the first performing "multi-head attention", the second being a simple feed-forward network. Throughout all layers, after each sub-layer, residual connections and layer normalization are applied, thus the intermediary output $\vec{o_L}$ after sub-layer L can be written as a function of the input $\vec{x_L}$, as $\vec{o_L} = LayerNorm(Sub_L(\vec{x_L}) + \vec{x_L})$.

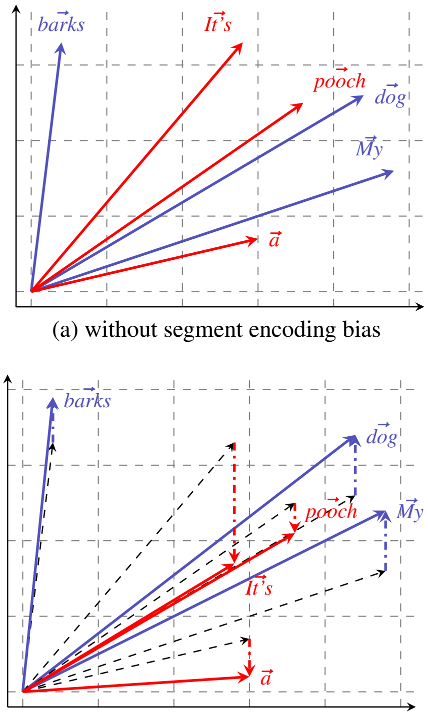
BERT is optimized on two training objectives. The first, called *masked language model*, is a variation on the Cloze test for reading proficiency (Taylor, 1953). The second, called *next sentence prediction* (NSP), corresponds to predicting whether two sentences are found one next to the other in the original corpus or not. Each example passed as input to BERT is comprised of two sentences, either contiguous sentences from a document, or randomly selected sentences. A special token `[SEP]` is used to indicate sentence boundaries, and the full sentence is prepended with a second special token `[CLS]` used to perform the actual prediction for NSP. Each token is transformed into an input vector using an input embedding matrix. To distinguish between tokens from the first and the second sentence, the model adds a learned feature vector $\vec{\mathtt{seg}}_A$ to all tokens from first sentences, and a distinct learned feature vector $\vec{\mathtt{seg}}_B$ to all tokens from second sentences; these feature vectors are called 'segment encodings'. Lastly, as Transformer models do not have an implicit representation of word-order, information regarding the index i of the token in the sentence is added using a positional encoding $p(i)$. Therefore, if the initial training example was "*My dog barks. It is a pooch.*", the actual input would correspond to the following sequence of vectors:

$$\begin{aligned}
&\vec{\mathtt{[CLS]}} + \vec{p(0)} + \vec{\mathtt{seg}}_A, \vec{My} + \vec{p(1)} + \vec{\mathtt{seg}}_A,\\
&\quad \vec{dog} + \vec{p(2)} + \vec{\mathtt{seg}}_A, \vec{barks} + \vec{p(3)} + \vec{\mathtt{seg}}_A,\\
&\quad\quad \vec{.} + \vec{p(4)} + \vec{\mathtt{seg}}_A, \vec{\mathtt{[SEP]}} + \vec{p(5)} + \vec{\mathtt{seg}}_A,\\
&\quad \vec{It} + \vec{p(6)} + \vec{\mathtt{seg}}_B, \vec{is} + \vec{p(7)} + \vec{\mathtt{seg}}_B,\\
&\quad \vec{a} + \vec{p(8)} + \vec{\mathtt{seg}}_B, \vec{pooch} + \vec{p(9)} + \vec{\mathtt{seg}}_B,\\
&\quad \vec{.} + \vec{p(10)} + \vec{\mathtt{seg}}_B, \vec{\mathtt{[SEP]}} + \vec{p(11)} + \vec{\mathtt{seg}}_B
\end{aligned}$$

Due to the general use of residual connections, marking the sentences using the segment encodings $\vec{\mathtt{seg}}_A$ and $\vec{\mathtt{seg}}_B$ can introduce a systematic offset within sentences. Consider that the first layer uses as input vectors corresponding to word, position, and sentence information: $\vec{w_i} + \vec{p(i)} + \vec{\mathtt{seg}}_i$; for simplicity, let $\vec{i_i} = \vec{w_i} + \vec{p(i)}$; we also ignore the rest of the input as it does not impact this reformulation. The output from the first sublayer $\vec{o_i^1}$ can be written:

$$\begin{aligned}
\vec{o_i^1} &= LayerNorm(Sub_1(\vec{i_i} + \vec{\mathtt{seg}}_i) + \vec{i_i} + \vec{\mathtt{seg}}_i)\\
&= \vec{b_l} + \vec{g^1} \odot \frac{1}{\sigma_i^1} Sub_1(\vec{i_i} + \vec{\mathtt{seg}}_i) + \vec{g^1} \odot \frac{1}{\sigma_i^1}\vec{i_i}\\
&\quad - \vec{g^1} \odot \frac{1}{\sigma_i^1}\mu(Sub_1(\vec{i_i} + \vec{\mathtt{seg}}_i) + \vec{i_i} + \vec{\mathtt{seg}}_i)\\
&\quad + \vec{g^1} \odot \frac{1}{\sigma_i^1}\vec{\mathtt{seg}}_i\\
&= \vec{\tilde{o}_i^1} + \vec{g^1} \odot \frac{1}{\sigma_i^1}\vec{\mathtt{seg}}_i
\end{aligned} \tag{2}$$

This equation is obtain by simply injecting the



(a) without segment encoding bias

(b) with segment encoding bias

Figure 2: Segment encoding bias

definition for layer-normalization.[3] Therefore, by recurrence, the final output $\vec{o_i^L}$ for a given token $\vec{w_i} + \vec{p(i)} + \vec{\texttt{seg}}_i$ can be written as:

$$\vec{o_i^L} = \vec{\tilde{o}_i^L} + \left(\bigodot_{l=1}^{L} \vec{g^l} \right) \odot \left(\prod_{l=1}^{L} \frac{1}{\sigma_i^l} \right) \times \vec{\texttt{seg}}_i \quad (3)$$

This rewriting trick shows that segment encodings are partially preserved in the output. All embeddings within a sentence contain a shift in a specific direction, determined only by the initial segment encoding and the learned gain parameters for layer normalization. In figure 2, we illustrate what this systematic shift might entail. Prior to the application of the segment encoding bias, the semantic space is structured by similarity ('*pooch*' is near '*dog*'); with the bias, we find a different set of characteristics: in our toy example, tokens are linearly separable by sentences.

[3]Layer normalization after sub-layer l is defined as:

$$\begin{aligned} \mathit{LayerNorm}_l(\vec{x}) &= \vec{b_l} + \frac{\vec{g_l} \odot (\vec{x} - \mu(\vec{x}))}{\sigma} \\ &= \vec{b_l} + \vec{g_l} \odot \frac{1}{\sigma}\vec{x} - \vec{g_l} \odot \frac{1}{\sigma}\mu(\vec{x}) \end{aligned}$$

where $\vec{b_l}$ is a bias, $\odot$ denotes element-wise multiplication, $\vec{g_l}$ is a "gain" parameter, σ is the standard deviation of components of $\vec{x}$ and $\mu(\vec{x}) = \langle \bar{x}, \ldots, \bar{x} \rangle$ is a vector with all components defined as the mean of components of $\vec{x}$.

4.2 Data & Experimental setup

If BERT properly describes a semantic vector space, we should, on average, observe no significant difference in token encoding imputable to the segment the token belongs to. For a given word type w, we may constitute two groups: $w_{\texttt{seg}_A}$, the set of tokens for this type w belonging to first sentences in the inputs, and $w_{\texttt{seg}_B}$, the set of tokens of w belonging to second sentences. If BERT counterbalances the segment encodings, random differences should cancel out, and therefore the mean of all tokens $w_{\texttt{seg}_A}$ should be equivalent to the mean of all tokens $w_{\texttt{seg}_B}$.

We used the same dataset as in section 3. This setting (where all paired input sentences are drawn from running text) allows us to focus on the effects of the segment encodings. We retrieved the output embeddings of the last BERT layer and grouped them per word type. To assess the consistency of a group of embeddings with respect to a purported reference, we used a mean of squared error (MSE): given a group of embeddings E and a reference vector $\vec{r}$, we computed how much each vector in E strayed from the reference $\vec{r}$. It is formally defined as:

$$\text{MSE}(E, \vec{r}) = \frac{1}{\#E} \sum_{\vec{v} \in E} \sum_{d} (\vec{v}_d - \vec{r}_d)^2 \quad (4)$$

This MSE can also be understood as the average squared distance to the reference $\vec{r}$. When $\vec{r} = \overline{E}$, i.e. $\vec{r}$ is set to be the average vector in E, the MSE measures variance of E via Euclidean distance. We then used the MSE function to construct pairs of observations: for each word type w, and for each segment encoding $\texttt{seg}_i$, we computed two scores: $\text{MSE}(w_{\texttt{seg}_i}, \overline{w_{\texttt{seg}_i}})$—which gives us an assessment of how coherent the set of embeddings $w_{\texttt{seg}_i}$ is with respect to the mean vector in that set—and $\text{MSE}(w_{\texttt{seg}_i}, \overline{w_{\texttt{seg}_j}})$—which assesses how coherent the same group of embeddings is with respect to the mean vector for the embeddings of the same type, but from the other segment $\texttt{seg}_j$. If no significant contrast between these two scores can be observed, then BERT counterbalances the segment encodings and is coherent across sentences.

4.3 Results & Discussion

We compared results using a paired Student's t-test, which highlighted a significant difference based on which segment types belonged to (p-value $< 2 \cdot 2^{-16}$); the effect size (Cohen's d =

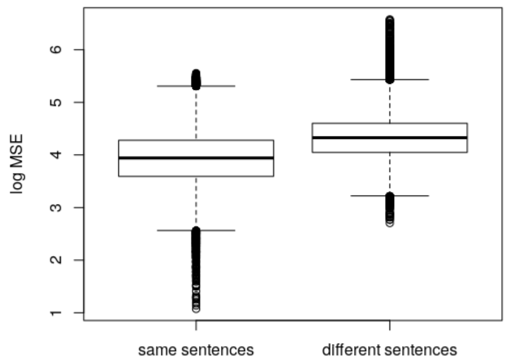


Figure 3: Log-scaled MSE per reference

-0.527) was found to be stronger than what we computed when assessing whether tokens cluster according to their types (cf. section 3). A visual representation of these results, log-scaled, is shown in figure 3. For all sets $w_{\texttt{seg}_i}$, the average embedding from the set itself was systematically a better fit than the average embedding from the paired set $w_{\texttt{seg}_j}$. We also noted that a small number of items yielded a disproportionate difference in MSE scores and that frequent word types had smaller differences in MSE scores: roughly speaking, very frequent items—punctuation signs, stopwords, frequent word suffixes—received embeddings that are *almost* coherent across sentences.

Although the observed positional effect of embeddings' inconsistency might be entirely due to segment encodings, additional factors might be at play. In particular, BERT uses absolute positional encoding vectors to order words within a sequence: the first word w_1 is marked with the positional encoding $p(1)$, the second word w_2 with $p(1)$, and so on until the last word, w_n, marked with $p(n)$. As these positional encodings are added to the word embeddings, the same remark made earlier on the impact of residual connections may apply to these positional encodings as well. Lastly, we also note that many downstream applications use a single segment encoding per input, and thus sidestep the caveat stressed here.

5 Experiment 3: Sentence-level structure

We have seen that BERT embeddings do not fully respect cross-sentence coherence; the same type receives somewhat different representations for occurrences in even and odd sentences. However, comparing tokens of the same type in consecutive sentences is not necessarily the main application of BERT and related models. Does the segment-based representational variance affect the structure of the semantic space, instantiated in similarities between tokens of different types? Here we investigate how segment encodings impact the relation between any two tokens in a given sentence.

5.1 Data & Experimental setup

Consistent with previous experiments, we used the same dataset (cf. section 3); in this experiment also mitigating the impact of the NSP objective was crucial. Sentences were thus passed two by two as input to the BERT model. As cosine has been traditionally used to quantify semantic similarity between words (Mikolov et al., 2013b; Levy and Goldberg, 2014a, e.g.), we then computed pairwise cosine of the tokens in each sentence. This allows us to reframe our assessment of whether lexical contrasts are coherent across sentences as a comparison of semantic dissimilarity across sentences. More formally, we compute the following set of cosine scores C_S for each sentence S:

$$C_S = \{\cos(\vec{v}, \vec{u}) \mid \vec{v} \neq \vec{u} \wedge \vec{v}, \vec{u} \in E_S\} \quad (5)$$

with E_S the set of embeddings for the sentence S. In this analysis, we compare the union of all sets of cosine scores for first sentences against the union of all sets of cosine scores for second sentences. To avoid asymmetry, we remove the `[CLS]` token (only present in first sentences), and as with previous experiments we neutralize the effects of the NSP objective by using only consecutive sentences as input.

5.2 Results & Discussion

We compared cosine scores for first and second sentences using a Wilcoxon rank sum test. We observed a significant effect, however small (Cohen's $d = 0.011$). This may perhaps be due to data idiosyncrasies, and indeed when comparing with a W2V (Mikolov et al., 2013a) trained on BooksCorpus (Zhu et al., 2015) using the same tokenization as BERT, we do observe a significant effect ($p < 0.05$). However the effect size is six times smaller ($d = 0.002$) than what we found for BERT representations; moreover, when varying the sample size (cf. figure 4), p-values for BERT representations drop much faster to statistical significance.

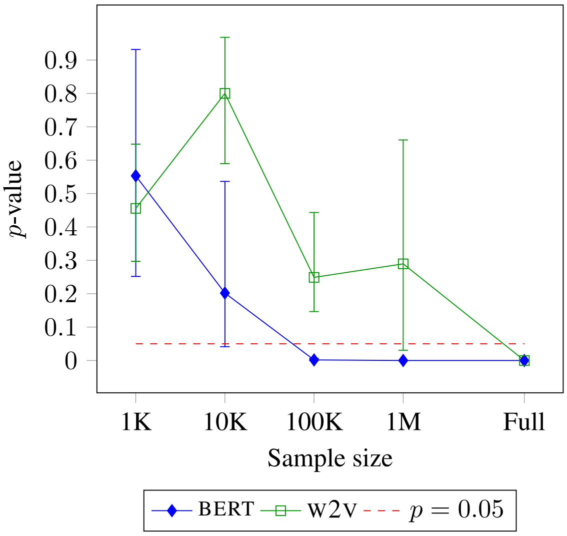


Figure 4: Wilcoxon tests, 1st vs. 2nd sentences

A possible reason for the larger discrepancy observed in BERT representations might be that BERT uses absolute positional encodings, i.e. the kth word of the input is encoded with $p(k)$. Therefore, although all first sentences of a given length l will be indexed with the same set of positional encodings $\{p(1), \ldots, p(l)\}$, only second sentences of a given length l preceded by first sentences of a given length j share the exact same set of positional encodings $\{p(j+1), \ldots, p(j+l)\}$. As highlighted previously, the residual connections ensure that the segment encodings were partially preserved in the output embedding: the same argument can be made for positional encodings. In any event, the fact is that we do observe on BERT representations an effect of segment on sentence-level structure. This effect is greater than one can blame on data idiosyncrasies, as verified by the comparison with a traditional DSM such as W2V. If we are to consider BERT as a DSM, we must do so at the cost of cross-sentence coherence.

The analysis above suggests that embeddings for tokens drawn from first sentences live in a different semantic space than tokens drawn from second sentences, i.e. that BERT contains two DSMs rather than one. If so, the comparison between two sentence-representations from a single input would be meaningless, or at least less coherent than the comparison of two sentence representations drawn from the same sentence position. To test this conjecture, we use two compositional semantics benchmarks: STS (Cer et al., 2017) and SICK-R (Marelli et al., 2014). These datasets are structured as triplets, grouping a pair

Model	STS cor.	SICK-R cor.
Skip-Thought	0.255 60	0.487 62
USE	0.666 86	0.689 97
InferSent	0.676 46	0.709 03
BERT, *2 sent. ipt.*	0.359 13	0.369 92
BERT, *1 sent. ipt.*	0.482 41	0.586 95
W2V	0.370 17	0.533 56

Table 1: Correlation (Spearman ρ) of cosine similarity and relatedness ratings on the STS and SICK-R benchmarks

of sentences with a human-annotated relatedness score. The original presentation of BERT (Devlin et al., 2018) did include a downstream application to these datasets, but employed a learned classifier, which obfuscates results (Wieting and Kiela, 2019; Cover, 1965; Hewitt and Liang, 2019). Hence we simply reduce the sequence of tokens within each sentence into a single vector by summing them, a simplistic yet robust semantic composition method. We then compute the Spearman correlation between the cosines of the two sum vectors and the sentence pair's relatedness score. We compare two setups: a "two sentences input" scheme (or *2 sent. ipt.* for short)—where we use the sequences of vectors obtained by passing the two sentences as a single input—and a "one sentence input" scheme (*1 sent. ipt.*)—using two distinct inputs of a single sentence each.

Results are reported in table 1; we also provide comparisons with three different sentence encoders and the aforementioned W2V model. As we had suspected, using sum vectors drawn from a two sentence input scheme single degrades performances below the W2V baseline. On the other hand, a one sentence input scheme seems to produce coherent sentence representations: in that scenario, BERT performs better than W2V and the older sentence encoder Skip-Thought (Kiros et al., 2015), but worse than the modern USE (Cer et al., 2018) and Infersent (Conneau et al., 2017). The comparison with W2V also shows that BERT representations over a coherent input are more likely to include some form of compositional knowledge than traditional DSMs; however it is difficult to decide whether some true form of compositionality is achieved by BERT or whether these performances are entirely a by-product of the positional encodings. In favor of the former, other research has suggested that Transformer-based architectures perform syntactic operations (Raganato and Tiedemann, 2018; Hewitt and Manning, 2019; Clark et al., 2019; Jawahar et al., 2019; Voita et al., 2019; Michel et al., 2019). In all, these results suggest that the semantic space of token representations from second sentences differ from that of embeddings from first sentences.

6 Conclusions

Our experiments have focused on testing to what extent similar words lie in similar regions of BERT's latent semantic space. Although we saw that type-level semantics seem to match our general expectations about DSMs, focusing on details leaves us with a much foggier picture.

The main issue stems from BERT's "next sentence prediction objective", which requires tokens to be marked according to which sentence they belong. This introduces a distinction between *first* and *second sentence of the input* that runs contrary to our expectations in terms of cross-sentence coherence. The validity of such a distinction for lexical semantics may be questioned, yet its effects can be measured. The primary assessment conducted in section 3 shows that token representations did tend to cluster naturally according to their types, yet a finer study detailed in section 4 highlights that tokens from distinct sentence positions (even vs. odd) tend to have different representations. This can seen as a direct consequence of BERT's architecture: residual connections, along with the use of specific vectors to encode sentence position, entail that tokens for a given sentence position are 'shifted' with respect to tokens for the other position. Encodings have a substantial effect on the structure of the semantic subspaces of the two sentences in BERT input. Our experiments demonstrate that assuming sameness of the semantic space across the two input sentences can lead to a significant performance drop in semantic textual similarity.

One way to overcome this violation of cross-sentence coherence would be to consider first and second sentences representations as belonging to distinct distributional semantic spaces. The fact that first sentences were shown to have on average higher pairwise cosines than second sentences can be partially explained by the use of absolute positional encodings in BERT representations. Although positional encodings are required so that the model does not devolve into a bag-of-word system, absolute encodings are not: other works have proposed alternative relative position encodings (Shaw et al., 2018; Dai et al., 2019, e.g.); replacing the former with the latter may alleviate the gap in lexical contrasts. Other related questions that we must leave to future works encompass testing on other BERT models such as the whole-words model, or that of Liu et al. (2019) which differs only by its training objectives, as well as other contextual embeddings architectures.

Our findings suggest that the formulation of the NSP objective of BERT obfuscates its relation to distributional semantics, by introducing a systematic distinction between first and second sentences which impacts the output embeddings. Similarly, other works (Lample and Conneau, 2019; Yang et al., 2019; Joshi et al., 2019; Liu et al., 2019) stress that the usefulness and pertinence of the NSP task were not obvious. These studies favored an empirical point of view; here, we have shown what sorts of caveats came along with such artificial distinctions from the perspective of a theory of lexical semantics. We hope that future research will extend and refine these findings, and further our understanding of the peculiarities of neural architectures as models of linguistic structure.

Acknowledgments

We thank Quentin Gliosca whose remarks have been extremely helpful to this work. We also thank Olivier Bonami as well as three anonymous reviewers for their thoughtful criticism. The work was supported by a public grant overseen by the French National Research Agency (ANR) as part of the "Investissements d'Avenir" program: Idex *Lorraine Université d'Excellence* (reference: ANR-15-IDEX-0004).

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Linear algebraic structure of word senses, with applications to polysemy. *CoRR*, abs/1601.03764.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry P. Vetrov. 2015. Breaking sticks and ambiguities with adaptive skip-gram. *CoRR*, abs/1502.07257.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Gemma Boleda. 2019. Distributional semantics and linguistic theory. *CoRR*, abs/1905.01896.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4349–4357. Curran Associates, Inc.

Olivier Bonami and Denis Paperno. 2018. A characterisation of the inflection-derivation opposition in a distributional vector space. *Lingua e Langaggio*.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, and Roger Wattenhofer. 2019. On the Validity of Self-Attention as Explanation in Transformer Models. *arXiv e-prints*, page arXiv:1908.04211.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. *arXiv e-prints*, page arXiv:1906.02715.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas M. Cover. 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electronic Computers*, 14(3):326–334.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *ACL*.

J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.

Gottlob Frege. 1892. Über Sinn und Bedeutung. *Zeitschrift fr Philosophie und philosophische Kritik*, 100:25–50.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. *arXiv e-prints*, page arXiv:1909.03368.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL-HLT*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Max M. Louwerse and Rolf A. Zwaan. 2009. Language encodes geographical information. *Cognitive Science 33 (2009) 5173*.

Marco Marelli and Marco Baroni. 2015. Affixation in semantic space: Modeling morpheme meanings with compositional distributional semantics. *Psychological review*, 122 3:485–515.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *NIPS*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? *arXiv e-prints*, page arXiv:1905.10650.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Belgium, Brussels. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

William Van Ormann Quine. 1960. *Word And Object*. MIT Press.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Siva Reddy, Ioannis P. Klapaftis, Diana McCarthy, and Suresh Manandhar. 2011. Dynamic and static prototype vectors for semantic composition. In *IJCNLP*.

Joseph Reisinger and Raymond Mooney. 2010. Multi-prototype vector-space models of word meaning. pages 109–117.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *CoRR*, abs/1507.01127.

Peter Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35, Belgium, Brussels. Association for Computational Linguistics.

Wilson Taylor. 1953. Cloze procedure: A new tool for measuring readability. *Journalism Quarterly*, 30:415–433.

Peter D. Turney and Patrick Pantel. 2010. From frequency and to meaning and vector space and models of semantics. In *Journal of Artificial Intelligence Research 37 (2010) 141-188 Submitted 10/09; published*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Lucas Vendramin, Pablo A. Jaskowiak, and Ricardo J. G. B. Campello. 2013. On the combination of relative clustering validity criteria. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, SSDBM, pages 4:1–4:12, New York, NY, USA. ACM.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation. In *Global Wordnet Conference*, Wroclaw, Poland.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Kellie Webster, Marta R. Costa-jussà, Christian Hardmeier, and Will Radford. 2019. Gendered ambiguous pronoun (GAP) shared task at the gender bias in NLP workshop 2019. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 1–7, Florence, Italy. Association for Computational Linguistics.

Matthijs Westera and Gemma Boleda. 2019. Don't blame distributional semantics if it can't do entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 120–133, Gothenburg, Sweden. Association for Computational Linguistics.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 887–898. Curran Associates, Inc.

Yukun Zhu, Jamie Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Curbing Feature Coding: Strictly Local Feature Assignment

Thomas Graf
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
mail@thomasgraf.net

Abstract

Graf (2017) warns that every syntactic formalism faces a severe overgeneration problem because of the hidden power of subcategorization. Any constraint definable in monadic second-order logic can be compiled into the category system so that it is indirectly enforced as part of subcategorization. Not only does this kind of feature coding deprive syntactic proposals of their empirical bite, it also undermines computational efforts to limit syntactic formalisms via subregular complexity. This paper presents a subregular solution to feature coding. Instead of features being a cheap resource that comes for free, features must be assigned by a transduction. In particular, category features must be assigned by an input strictly local (ISL) tree-to-tree transduction, defined here for the first time. The restriction to ISL transductions correctly rules out various deviant category systems.

1 Introduction

Theoretical and computational linguists both strive to identify limited models of language that furnish sufficient power without allowing for excessive overgeneration. Recently, Graf (2017) noted that the findings of Graf (2011) and Kobele (2011) point towards a major loop hole in all current theories of syntax. The category system can be abused to encode additional information about the syntactic tree, and the usual subcategorization requirements can then be used to enforce a certain kind of synchronization between parts of the tree. For instance, the category DP may be split into DP[+NPI] and DP[−NPI] depending on whether the DP is an NPI, and the category X of each selecting head becomes X[+NPI] if the argument it selects contains an unlicensed NPI. This simple strategy has been known for a long time but did not raise serious concerns as it is widely accepted that all grammar formalisms "leak" in the sense that they also allow for some unnatural patterns.

But the extent of the problem for linguistic theory has not been fully appreciated. Graf (2017) shows how this strategy can be generalized to flout all island constraints, enforce constraints that lack any notion of locality, and even add highly unnatural counting requirements to the grammar. Every constraint that can be defined in monadic second-order logic is expressible through category refinement. This allows for very unnatural constraints, e.g. enforcing verb-second word order iff the sentence contains exactly three relative clauses or both a Principle A violation and a word in which unbounded tone plateauing is not obeyed. The only way to preclude this is to restrict the shape of category systems, but Graf (2017) argues that the usual linguistic requirements on syntactic categories are insufficient. Hence every syntactic formalism lacks a key mechanism to distinguish natural patterns from unnatural ones, resulting in massive overgeneration.

This paper proposes a computational solution to this problem, drawing from recent work on subregular complexity. Features no longer come part and parcel with lexical items, but must be assigned to tree structures by a transduction. An unnatural feature system that keeps track of, say, a counting dependency, requires a very powerful transduction. The category systems of natural languages, on the other hand, can be handled by much simpler means. I argue that these category systems only require inspection of a lexical item's local context. This intuition is formalized by generalizing the input-strictly local (ISL) string-to-string mappings of Chandlee (2014) to ISL tree-to-tree transductions. To the best of my knowledge, this is the first time a subregular tranduction class is defined for trees, and I hope it will be a fertile vantage point for mathematical and empirical work alike.

The paper deviates slightly from the usual structure. Since the problem is also of interest to theoretical linguists and the proposed solution is fairly intuitive, the first half focuses on the big picture and keeps formal concepts to a minimum (§2). The mathematical aspects are then worked out in §3, the most important of which is the formal definition of ISL tree transductions (§3.2).

2 Problem and Solution: Informal Sketch

The power of category systems and subcategorization is best illustrated with an example (§2.1). This makes it clear what unnatural category systems may look like, and in what respects they clearly differ from natural ones (§2.2). The problem of category abuse in syntax is actually an instance of the more general phenomenon of feature coding, which also appears in the domain of subregular complexity (§2.3). But subregular complexity also provides a way of measuring the complexity of feature systems via transductions. With strict limits on the power of these transductions, many of the unnatural category systems are correctly ruled out (§2.4) while it becomes possible to formulate new syntactic universals (§2.5).

2.1 A Grammar with Odd/Even Counting

Let us start with a toy example from Minimalist grammars (MGs; Stabler, 1997, 2011) that illustrates the power of syntactic categories. MGs are closely modeled after Minimalist syntax, and subcategorization is encoded via category and selector features that drive the operation *Merge*. A head with *selector feature* X^{+} can only be merged with a phrase whose head has *category feature* X^{-}. This matching of features is called *feature checking*. The category feature of a lexical item l can only be checked once all selector features of l have been checked. While exceedingly simple, this system is already too powerful as a model of subcategorization in natural languages.

Consider the MG G where the only pronounced lexical items are *foo* and *bar*, which may have the category features E^{-} or O^{-}. By default, the category feature is O^{-}. But if the lexical item carries a selector feature O^{+} or E^{+}, the category feature must be the opposite of that selector feature (E^{-} or O^{-}, respectively). Hence *foo* and *bar* may have the feature strings O^{-}, $\mathrm{E}^{+}\mathrm{O}^{-}$, or $\mathrm{O}^{+}\mathrm{E}^{-}$. Besides *foo* and *bar*, the MG only has an unpronounced C-

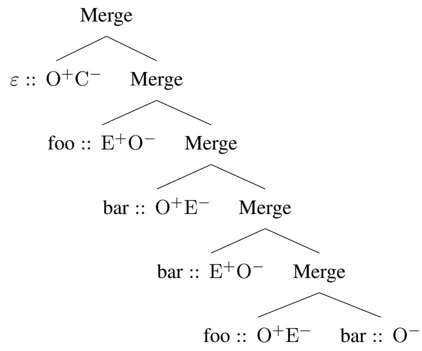


Figure 1: Derivation tree for *foo bar bar foo bar*

head, which must always be the last lexical item to be merged. The C-head carries the selector feature O^{+}. Overall, G consists of the following lexical items:

(1) **MG G with even/odd alternation**

ε :: $\mathrm{O}^{+}\mathrm{C}^{-}$	foo :: $\mathrm{E}^{+}\mathrm{O}^{-}$	foo :: O^{-}
	foo :: $\mathrm{O}^{+}\mathrm{E}^{-}$	
	bar :: $\mathrm{E}^{+}\mathrm{O}^{-}$	bar :: O^{-}
	bar :: $\mathrm{O}^{+}\mathrm{E}^{-}$	

The MG generates any string over *foo* and *bar* whose length is odd. The reasoning for this is as follows: the derivation must start with either foo :: O^{-} or bar :: O^{-}. From this point on, selecting heads alternate between E^{-} and O^{-}, but only a head carrying O^{-} can be selected by the C-head to end the derivation. The end result is that the number of pronounced lexical items in the tree must be odd, as is also illustrated in Fig. 1. The MG above thus instantiates a simple case of *modulo* counting at the string level.

2.2 (Un)Naturalness of the Example MG

The example grammar G in (1) is highly unnatural in several respects. First of all, string length does not seem to be a relevant criterion for natural language syntax. This definitely holds for *modulo* counting, which is unheard of. But even absolute size requirements are hard to come by unless one abandons the well-motivated competence-performance distinctions. A potential counterexample is Heavy NP-shift, which is sensitive to a constituent's size and thus, possibly, its string length. But even here processing provides a more plausible explanation (cf. Liu, 2018). Syntax itself seems to be completely blind to size, be it string length or the size of a tree.

Perhaps even more important is the fact that O^{-} and E^{-} do not convey intrinsic information of the lexical item l that carries them. Instead, these categories represent properties of the whole subtree. Hence the category is highly context-dependent. If one wanted to insert another instance of *foo* or *bar* in the subtree headed by l, one would also have to change the category of l because of how O^- and E^- have to alternate. The change of l's category then requires changing the category of l's selector, the selector of l's selector, and so on. This directly contradicts a basic principle of selection: a lexical item selects for its argument, not the argument(s) of its argument. A verb selecting a PP may restrict the shape of the P-head, but not the DP inside the PP. And no lexical item can freely select any head of any category as long as the selected subtree satisfies some other property. Subcategorization enforces head-head dependencies, not head-subtree dependencies, and any category system that allows the latter to be reduced to the former is missing a key aspect of natural language.

2.3 The Full Extent of the Problem

As was already mentioned in the introduction, the example above is but the tip of the iceberg. Without restrictions on the category system, any arbitrary constraint can be enforced as long as it is definable in monadic second-order logic. Graf (2017, p. 22–24, p. 27f) gives several illustrative examples of overgeneration and explains in detail why the usual heuristics (e.g. syntactic distribution, morphological inflection) are not sufficient to distinguish natural from unnatural category systems. Beyond *modulo* counting, this kind of *feature coding* also allows for, among other things, strange constraint interactions ("Satisfy either verb-second or Principle A, but not both"), symmetric counterparts of existing constraints (Reverse Principle A: every reflexive must c-command a suitable R-expression), and displacement mechanisms that do not use movement and hence bypass island constraints. All of this becomes possible because feature coding abuses categories as a local buffer for non-local information, erasing all locality and complexity differences between constraints.

The potential abuse of syntactic categories is actually an instance of a more general problem that has to be carefully avoided in subregular phonology. Subregular phonology (see Heinz 2018 and references therein) has identified very restricted subclasses of the regular string languages that still furnish enough power for phonology. Crucially, though, these claims depend on the choice of features because every regular pattern can be made subregular by introducing additional features. In formal terms: every recognizable set is a projection of a local set (cf. Rogers, 1997).

For instance, the regular string language of odd-length strings over a can be pushed into the extremely weak subclass of *strictly 2-local* string languages if one introduces a feature [±odd]. A string like $a\ a\ a\ a\ a$ would then be represented as a[+odd] a[−odd] a[+odd] a[−odd] a[+odd]. The language with the diacritic [±odd] feature is strictly 2-local because it can be expressed in terms of constraints that involve at most two segments:

(2) **Strictly 2-local constraints**

a. Every string must start with a[+odd] and end with a[+odd].

b. a[+odd] must not follow a[+odd].

c. a[−odd] must not follow a[−odd].

The example in §2.1 is a syntactic analog of this trick, with O^- and E^- filling the roles of [±odd]. In all these cases, feature coding obfuscates subregular complexity by precompiling complex dependencies into an invisible alphabet of features and diacritics.

The feature coding problem is less severe in subregular phonology thanks to the restriction to articulatory features, which can usually be replaced by the actual segments without changing anything substantive about the analysis.[1] In syntax, features play a much more vital role as two representations may look exactly the same except for their feature make-up.

For instance, Fig. 2 gives an MG dependency tree representation for *the gardeners water their flowers*, while adding the movement features top^- to *their* and top^+ to *water* yields the MG dependency tree representation of the very different topicalization sentence *their flowers, the gardeners water*. The movement features are an essential part of the representation. Similarly, category and selector features can be crucial for head-argument relations in MG derivation trees. In Fig. 1, switching the feature strings of the bottom-most *foo* and

[1] One notable exception is Baek (2018). She adds a limited number of structural features to define a subregular class that lies strictly between the classes TSL (Heinz et al., 2011) and ITSL (De Santo and Graf, 2019).

bar would yield a new string *foo bar bar bar foo*. This is because derivation trees encode head-argument relations only via Merge features, not via dominance or linear order. It is not surprising, then, that all the recent work extending the subregular perspective from phonology to syntax relies on feature in one way or another (Graf, 2018; Graf and Shafiei, 2019; Graf and De Santo, 2019; Vu, 2018; Vu et al., 2019).

But even if features could be done away with, that would be too extreme a step as they can still be useful. Consider once more the case of topicalization movement. This involves three computational steps: I) identifying the mover and the target site, II) determining whether topicalization movement is licit, and III) displacing the topicalized phrase. Without features, the first two steps would have to be handled by the same computational device, which first makes a non-deterministic choice as to what should move where, and then decides whether this instance of movement obeys all relevant constraints. By making features an integral part of the representation, we factor out the first step in order to isolate the complexity of the second step. But without a restrictive theory of features, there is the risk of factoring out more than intended. This would lead to misleading claims about subregular complexity that are merely artifacts of feature coding. Subregular syntax thus finds itself in a precarious situation where the very thing it depends on also threatens to undermine all its findings.

The original problem of syntactic categories thus is but a piece of the larger puzzle of how to avoid feature coding. The brute force solution of shunning features altogether is not workable in syntax. Features distinguish otherwise identical representations; theoretical and computational linguists alike are too accustomed to thinking in terms of features; and features do allow for insightful factorizations of complexity. The problem is not features as such, it is the lack of a measuring rod for how much complexity has been shifted into the feature system.

2.4 Solution: Strictly Local Feature Assignment

Features come for free under current models of complexity because they are representational devices. Subregular complexity takes the representations for granted and then investigates how hard a

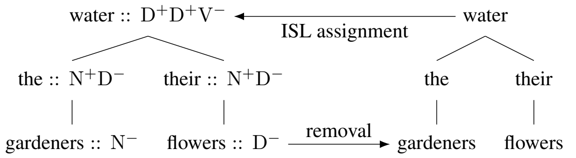


Figure 2: Feature assignment as a transduction problem between a feature-annotated MG dependency tree (left) and its feature-free counterpart (right)

given dependency would be to enforce over these representations. In order to assess the complexity of feature systems, we have to decouple them from the representations. Intuitively speaking, we want to measure the complexity of constructing a feature-annotated representation from its feature-free counterpart.

Formally, this takes the shape of a transduction problem. For strings, transductions are a formal counterpart of rewrite rules, and for trees they are similar to syntactic transformations in the sense of Chomsky (1965). Chandlee (2014) defines a particularly weak kind of string transductions known as *input strictly local* (ISL). An ISL transduction considers only the local context of a symbol when deciding how it should be rewritten. Word-final devoicing and intervocalic voicing are examples of ISL transductions in phonology, whereas long-distance sibilant harmony would not be ISL because the rewriting of a sibilant can depend on other segments that are arbitrarily far away. ISL can be lifted from strings to trees: a node in a tree may be rewritten in various ways depending on its local context in the tree. A transduction is ISL-k iff all local contexts can be limited to at most k levels (a mother-daughter configuration, for example, involves two levels).

Figure 2 illustrates the approach with a feature-annotated MG dependency tree for *the gardeners water their flowers*. The question at hand is whether the familiar categories D, N, and V can be assigned by an ISL transduction. We take a feature-annotated representation like the one of the left and remove all category and selector features. Then we have to define an ISL transduction that takes us back to the original representation. If this can be done with any well-formed tree, then the whole feature system is *ISL recoverable*.

For the specific tree in Fig. 2, we need an ISL-2 transduction. The feature annotations for *the*, *their*, and *gardeners* can be recovered without any further context information just from the phonetic exponents. That is the case because there simply are no alternative feature annotations for these lexical items in English. With *water* and *flowers*, on the other hand, there is ambiguity as each one of them could be either a noun or a verb. But in both cases a minimum amount of context is sufficient to disambiguate their categories. Since *flowers* is selected by *the*, which can only be a determiner, *flowers* must be a noun. Similarly, *water* must be a verb because it selects *the* and *their*, neither one of which could be an argument of the noun *water*. Inspecting the daughters or the mother of a node requires a context with two levels, so the transduction is ISL-2 for this specific example. The complexity of the whole feature system corresponds to the weakest transduction that works for all well-formed trees (usually there will be infinitely many of those; therefore, conclusive complexity results require proofs rather than examples).

The feature system of the MG G in Sec. 2.1 is not ISL recoverable. This follows from the fact that it is not ISL-k recoverable for any $k \geq 1$. For the sake of simplicity, we will once again use a dependency tree format as in Fig. 2 instead of the derivation tree format in Fig. 1. Now suppose that the features for the left tree in Fig. 3 could be correctly assigned from the middle tree by an ISL-k transduction. Since the transduction is ISL-k, the features assigned to *foo* depend exclusively on some context with at most k levels. Crucially, *foo* will always receive the same features as long as the context remains the same. But now compare this to the tree on the right. Here *foo* has switched positions with *bar* below it, inducing a change in its feature make-up. Yet the locally bounded context for *foo* has not changed at all — the middle tree could also be a description for the right tree depending on the values of $m' \geq k$ and $n' \geq k$. Hence the feature annotation for *foo* varies despite identical contexts, which proves that the feature system is not ISL recoverable. In fact, no ISL transduction can handle any feature system that involves *modulo* counting.

2.5 Some Linguistic Implications

ISL recoverability correctly rules out some of the most egregious patterns and constraints. But we can try to further limit feature systems based on the size of contexts. Instead of ISL recoverability, the relevant restriction would be ISL-k recoverability for some small k.

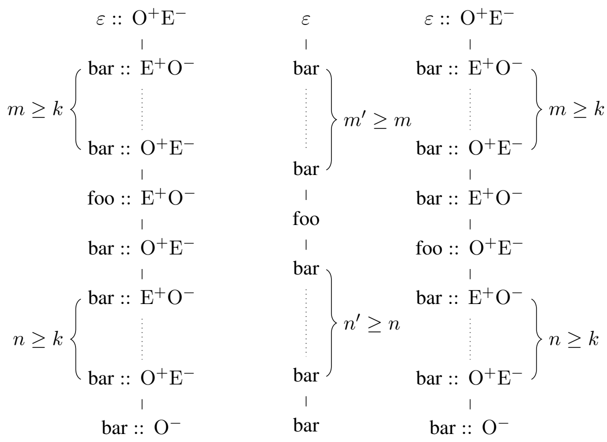


Figure 3: Modulo systems are not ISL-k recoverable

Note first that the value of k can vary depending on other assumptions. MG derivation trees like the one in Fig. 1 display a greater distance between heads and arguments than MG dependency trees like the one in Fig. 2, so the latter will minimize the value for k. This does not mean that the latter is linguistically preferable, but rather that k cannot be fixed independently of the choice of representation. The value of k will also depend greatly on the shape of the phonetic exponents. Fully inflected forms can provide crucial clues about a lexical item's category that would be missing from the uninflected roots postulated in Distributed morphology (Halle and Marantz, 1993). It remains to be seen which set of assumptions and parameters will prove most insightful.

At this point, though, I put forward a maximally restrictive conjecture. Based on a preliminary survey of English data and the linguistic *bon mot* that heads do not select for arguments of arguments, I contend that the category systems of natural languages are maximally simple:

(3) **Complexity of category systems**
Given MG dependency trees with uninflected roots as exponents (e.g. $\sqrt{\text{destroy}}$, $\sqrt{\text{water}}$), it holds for every natural language that all its category and selector features are ISL-2 recoverable.

The conjecture in (3) predicts that whenever a lexical item is categorially ambiguous, its category feature can be determined by inspecting the selecting head or the heads of the selected arguments. Even if ISL-2 recoverability ultimately turns out to be too strong an assumption, ISL-k recoverability still rules out many undesirable feature systems and reins in feature coding while allowing for limited categorial ambiguity.

ISL recoverability also has some more indirect consequences. One prediction is that no natural language can have an arbitrarily long sequence $x_1, \ldots, x_n$ such that I) each x_i is an empty head, and II) x_i selects x_{i+1} and nothing else ($1 \leq i < n$). This prediction follows from the fact that unpronounced lexical items provide no overt clues about their category. If the local context does not furnish any pronounced material, local category inference hinges on structural differences. Since the configuration above is structurally uniform, there is insufficient information to correctly infer the categories of all empty heads. This case is interesting because of the proliferation of empty heads in Minimalist syntax. If there is any clear counterexample to (3), it is likely to involve empty heads.

It should also be noted that ISL recoverability is only expected to hold for category and selector features. Features that participate in long-distance dependencies like movement cannot be reliably assigned by an ISL transduction.[2] Consider once more our topicalization example from before. Whether *water* should receive a top^+ feature to license topicalization depends on whether there is some head with a matching top^- feature to undergo topicalization. In the case at hand, this can be made based on the local context alone. But in general, a mover can be arbitrarily far away from its target site, as in *this author, John thinks that Bill said that Mary really adores*. Correct assignment of top^+ thus requires a context of unbounded size, which is impossible with ISL transductions.

Many empirical and theoretical issues remain to be settled. The MG corpus of Torr (2017) may provide valuable clues about the feasibility of conjecture (3), but it must be supplemented by a broad range of typological data. On the formal side, studying the recoverability of movement features will require more powerful extensions of ISL tree transductions. The next section fully formalizes ISL transductions to provide a suitable vantage point for this future work.

[2] In grammars with adjunction, subcategorization can also become a long-distance dependency depending on one's choice of representation (Graf, 2018). A modified version of (3) would predict that the uninflected root of each adjunct still provides enough information to reliably infer category and selector features. I am much more skeptical that this will turn out to be true across all languages.

3 Formal Definitions

This section puts the informal discussion of the preceding section on a formal footing by defining ISL recoverability in terms of *ISL tree relabelings*. But in order to simplify future work on feature recoverability, I define the more general class of ISL tree transductions, which ISL tree relabelings are a particular simple subtype of. The definition of ISL tree transductions differs markedly from that of other tree transductions. Building on Gorn domains and tree contexts (§3.1), I define an ISL tree transducer as a finite set of triples, each one of which maps a node n to a tree context based on the configuration n appears in. The ISL transduction then combines all these tree contexts to yield the final output tree (§3.2). Given this formal apparatus, feature recoverability is easy to state in rigorous terms (§3.3).

3.1 Technical Preliminaries

We define trees as finite, labeled Gorn domains (Gorn, 1967). First note, though, that we use $\mathbb{N}$ to denote the set of all positive natural numbers, i.e. $\{1, 2, 3, \ldots\}$ rather than $\{0, 1, 2, 3, \ldots\}$ — this is non-standard, but will slightly simplify the usage of indices in the definition of ISL-k transducers.

A *Gorn domain* D is a set of strings drawn from $\mathbb{N}^*$, which are called *(Gorn) addresses*, or simply *nodes*. Every Gorn domain must satisfy two closure properties: for all $u \in \mathbb{N}^*$ and $1 \leq i \leq j$ it holds that $uj \in D$ implies both $u \in D$ and $ui \in D$. This entails the inclusion of the empty string ε, which denotes the root. Addresses are interpreted such that u immediately dominates each ui, and each ui is the immediate left sibling of $u(i+1)$.

A Σ*-tree* is a pair $t := \langle D, \ell \rangle$ where D is a finite Gorn domain and $\ell : D \to \Sigma$ is a total function that maps each address to its label, i.e. a member of the alphabet Σ. The *depth* of t is equivalent to the length of the longest Gorn address.

A (Σ, n)*-context* is a Σ-tree whose leaf nodes may also have labels drawn from the set $\{\square_1, \ldots, \square_n\}$ of *ports*, which must be disjoint from Σ. Suppose we are given a (Σ, n)-context $c := \langle D_c, \ell_c \rangle$ with $m \leq n$ ports labeled $\square_i$ at addresses $a_1, \ldots, a_m$, as well as a tree (or context) $s := \langle D_s, \ell_s \rangle$. Then we use $c[\square_i \leftarrow s]$ to denote the result of substituting s for each $\square_i$ in c. This is a new tree $t := \langle D, \ell \rangle$ such that

- $D := D_c \cup \{a_j d \mid 1 \leq j \leq m, d \in D_s\}$, and

- for every $b \in D$

$$\ell(b) := \begin{cases} \ell_s(d) & \text{if } b = a_j d \\ & (1 \leq j \leq m,\ d \in D_s) \\ \ell_c(b) & \text{otherwise} \end{cases}$$

The construction also generalizes to multiple simultaneous substitutions, as in $c[\Box_i \leftarrow s, \Box_j \leftarrow t]$. If c contains no node labeled $\Box_i$, then $c[\Box_i \leftarrow s, \Box_j \leftarrow t] = c[\Box_j \leftarrow t]$ (and $c[\,] = c$).

If S is a set, then substitution can apply in two ways. With *synchronous substitution*, $t[\Box_i \leftarrow S] := \{t[\Box_i \leftarrow s] \mid s \in S\}$. *Asynchronous substitution*, denoted $t[\Box_i \leftharpoondown S]$, yields $\{t[\Box_{i_1} \leftarrow s_1, \ldots, \Box_{i_n} \leftarrow s_n] \mid s_1, \ldots, s_n \in S\}$, assuming that t contains exactly n occurrences of $\Box_i$. Substitution with sets and multiple simultaneous substitutions will be crucial for ISL transductions.

3.2 ISL Transductions

Chandlee (2014) defines ISL string-to-string mappings in terms of deterministic, finite-state string-to-string transducers. Even though the definition does not provide an explicit look-ahead component, ISL mappings can emulate finitely bounded look-ahead via a delayed-output strategy. Suppose, for instance, that a is rewritten as b before d, as c before e, and just as a before f. This is emulated by deleting a and rewriting the next symbol as either bd, ce, or af. Later works define ISL functions in terms of local contexts (not to be confused with (Σ, n)-contexts), and those definitions make look-ahead a standard component to simplify practical work (Chandlee and Heinz, 2018; Graf and Mayer, 2018; De Santo and Graf, 2019).

With tree transducers, the emulation of finitely bounded look-ahead is a much more complex affair that depends on various parameters such as directionality (top-down or bottom-up), totality, and determinism. For this reason, I explicitly add finite look-ahead in the subsequent definitions. I will also allow for non-determinism as future work may require transductions than can handle optionality (e.g. whether a node should receive a movement feature to undergo topicalization).

For the sake of generality and as a starting point for future work, I first define a version of ISL tree transductions that allows for non-determinism, deletion, and copying, and that can run in two different modes of operation (*synchronous* or *asynchronous*). This is subsequently limited to the special case of ISL relabelings, which are the formal core of feature recoverability.

Definition 1 (ISL tree transducer). For any $k \geq 1$, an *ISL-k tree transducer* from Σ-trees to Ω-trees is a finite set τ of *ISL-k rewrite rules* $\langle s, a, t\rangle$, where

- s is a Σ-tree of depth $i < k$,
- a is a node (i.e. a Gorn address) of s with $d \geq 0$ daughters,
- and t is an (Ω, d)-context. ⌟

Definition 2 (Synchronous ISL transduction). The transduction realized by an ISL-k transducer in *synchronous mode* is defined in a recursive fashion. First, a node b in tree u can be *targeted* by an ISL-k context $\langle s, a, t\rangle$ iff there is some $p \in \mathbb{N}^*$ such that

node match $b = pa$, and

label match for all nodes g of s, $\ell_s(g) = \ell_u(pg)$,

full-width match for all nodes gi of s with $g \in \mathbb{N}^*$ and $i \in \mathbb{N}$, if pgj is a node of u ($j > i$), then gj is a node of s.

Now suppose furthermore that n in u has $d \geq 0$ daughters. Given an ISL-k tree transducer τ, we use $\overleftarrow{\overleftarrow{\tau}}(u, b)$ to denote the set of all trees $t[\Box_1 \leftarrow \overleftarrow{\overleftarrow{\tau}}(u, b1), \ldots, \Box_d \leftarrow \overleftarrow{\overleftarrow{\tau}}(u, bd)]$ such that there is a rewrite rule $\langle s, a, t\rangle$ in τ that targets node b in u. If this set is empty, $\overleftarrow{\overleftarrow{\tau}}(u, b)$ is undefined. For any Σ-tree t, we may simply write $\overleftarrow{\overleftarrow{\tau}}(t)$ instead of $\overleftarrow{\overleftarrow{\tau}}(t, \varepsilon)$. For any tree language L, the transduction computed by τ in *synchronous mode* is $\overleftarrow{\overleftarrow{\tau}}(L) := \{\langle i, o\rangle \mid i \in L, o \in \overleftarrow{\overleftarrow{\tau}}(i)\}$. A transduction is *synchronous input strictly k-local* (sISL-k) iff it can be computed by some ISL-k transducer in synchronous mode. It is *synchronous input strictly local* (sISL) iff it is sISL-k for some $k \geq 1$. ⌟

The definition of *asynchronous input strictly k-local* (aISL-k) transductions is exactly the same, except that $\overleftarrow{\overleftarrow{\tau}}$ is replaced by $\overleftharpoondown{\tau}$ such that $\overleftharpoondown{\tau}(u, b)$ denotes the set $t[\Box_1 \leftharpoondown \overleftharpoondown{\tau}(u, b1), \ldots, \Box_d \leftharpoondown \overleftharpoondown{\tau}(u, bd)]$. ISL is used as a shorthand for sISL or aISL, ignoring transduction mode.

The definition of ISL transductions differs from that of other tree transductions in that the input tree is not altered incrementally to yield the output tree. Instead, each node in the input contributes a context to the output, or rather, a range of possible contexts in the case of a non-deterministic transduction. The transduction then stitches these contexts together in order to arrive at a single tree structure. This stitching is accomplished by the recursive step of mapping $\tau(u, b)$ to $t[\Box_1 \leftarrow \tau(u, b1), \ldots, \Box_d \leftarrow \tau(u, bd)]$. Each $\tau(u, bi)$ ($1 \leq i \leq d$) corresponds to a context produced from the i-th daughter of the node b in u, and these contexts are inserted into the appropriate ports of the context t produced from n. If n is a leaf node, its output structure is a tree instead of a context. This ensures that the recursion step terminates eventually.

Example. Figure 4 specifies a fragment of an ISL-3 transducer for translating multiplication trees to addition trees (assuming no numbers larger than 3). For simplicity, the ISL rewrite rules are written in a context-free format with a box around the node to be rewritten. An underscore is used to match any arbitrary node label. On the right, a particular input-output mapping is shown with the transducer running in asynchronous mode. In synchronous mode, all $\Box_i$ in the output of rule G would have to be replaced by the same tree. ⌟

It is easy to see that every ISL-k string transduction is an ISL-k tree transduction over unary branching trees. This shows that ISL-k tree transducers are a natural generalization of ISL for strings. However, the current definition goes far beyond ISL string mappings in that it allows for non-determinism and copying.

Definition 3 (Transducer subtypes). An ISL-k tree transducer τ is

deterministic iff it holds for every Σ-tree u that no node of u can be targeted by more than one context of τ,

linear/non-deleting iff all contexts $\langle s, a, t\rangle$ of τ are such that if the node at address a in s has $d \geq 1$ daughters, then t contains every port $\Box_i$ at most once/at least once ($1 \leq i \leq d$),

structure preserving iff all rewrite rules $\langle s, a, t\rangle$ of τ are such that t is of the form $\omega(\Box_1, \ldots, \Box_d)$ ($\omega \in \Omega$).

A deterministic, structure preserving ISL-k transducer is called an *ISL-k relabeling*. ⌟

A structure preserving ISL transducer never changes the structure of the input tree. Structure preservation thus entails linearity, which is why the latter is not mentioned in the definition of relabelings. Linearity in turn removes the distinction between synchronous and asynchronous mode as no $\Box_i$ ever has more than one occurrence. Only this very limited type of ISL transducers is relevant for feature recoverability.

3.3 ISL Feature Recoverability

We are finally in a position to define the notion of ISL recoverability that was informally discussed in §2.4. In order to clearly separate features from other parts of the alphabet, we have to track them in a separate component. MGs make this split fully explicit, with a lexical item's phonetic exponent a member of Σ and their feature annotation a string over an entirely separate set of features. Other formalisms such as TAG or GPSG can also be recast along these lines.

Definition 4. Let F be a set of features. An *F-annotated Σ-tree* is a tree whose labels are drawn from $\Sigma \times F^*$. ⌟

Definition 5. Let F be a set of features and e a function that maps each $\langle \sigma, f\rangle \in \Sigma \times F^*$ to σ. Then F is *ISL-k recoverable* with respect to language L of F-annotated Σ-trees iff there is an ISL-k transducer τ such that $\tau(e(t)) = t$ for all $t \in L$. ⌟

Note that feature recoverability can vary depending on the particulars of the tree languages. A feature that may not be recoverable with respect to L may be recoverable with respect to L'. Consider once more the grammar in §2.1. If *foo* always had to carry O^-, and *bar* always had to carry E^-, then those category features would be recoverable even though they still encode an even/odd alternation. In this hypothetical scenario, the alternation is tied to overt exponents, reducing *modulo* counting to a strictly 2-local alternation of lexical items. In the other direction, even the simplest (non-trivial) category system cannot be recovered from a language where all lexical items are unpronounced. And as a reviewer correctly points out, if one puts no restrictions on the use of empty heads, features can be encoded in terms of specific structural configurations with empty heads. Feature recoverability thus is a fluid notion that depends equally on the nature of Σ, the syntactic assumptions about structure and phonetic exponents, and the overall complexity of the tree language.

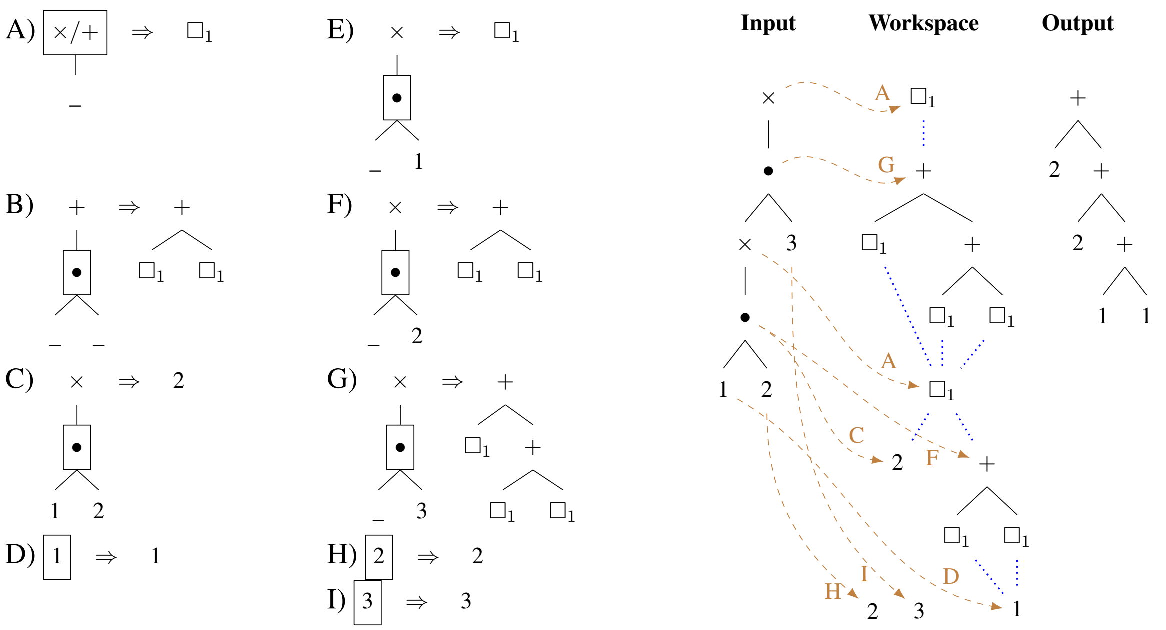


Figure 4: A non-deterministic ISL-3 tree transducer (left) for converting a tree with addition and multiplication to addition only (right). The workspace depicts I) how each node is rewritten as one or more contexts, and II) all possible options for combining these contexts into a particular output tree via substitution steps that are based on the structure of the input tree. Dashed arrows are annotated with the corresponding rewrite rules. The transducer is assumed to operate in asynchronous mode. The output displays one out of $2^3 = 8$ options that differ in when and where rewrite rules C and F are used. In synchronous mode, there would be only two distinct outputs.

4 Conclusion

ISL tree transductions — or more precisely, ISL tree relabelings — offer a reasonable approximation of the limits of category systems in natural languages. I conjecture that all natural languages are such that the category of a lexical item can be inferred from its local context in a tree without any feature annotations. In combination with standard assumptions about linguistic structure, feature recoverability is a powerful restriction that eliminates many of the undesirable cases of feature coding identified in Graf (2017). It also makes strong empirical predictions that merit further investigation by linguists.

Many questions had to remain open. On the formal side, this includes abstract characterizations as well as core properties of ISL tree transductions, e.g. (non-)closure under intersection, union, and composition. The relations to other transduction classes are largely unknown. I conjecture that (deterministic) synchronous/asynchronous ISL transductions are subsumed by (deterministic) bottom-up/top-down transductions with finite look-ahead. Linear ISL transductions should be subsumed by both. The movement features of MGs will require a more powerful kind of transduction, possibly based on the string class TSL (Heinz et al., 2011). There also seems to be a deep connection between feature recoverability and the notion of inessential features (Kracht, 1997; Tiede, 2008).

From a linguistic perspective, one pressing question is to what extent feature recoverability depends on whether syntax uses fully inflected lexical forms or underspecified roots. If fully inflected lexical items do not reduce the complexity of the ISL transduction, or allows for unnatural constraints that would not be possible otherwise, that would be a powerful argument that syntax indeed has no need for anything beyond simple roots.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. This paper benefited greatly from the feedback of Jeffrey Heinz, Dakotah Lambert, and three anonymous reviewers. I am indebted to the participants of the University of Tromsø's workshop *Thirty Million Theories of Syntactic Features*, which lit the initial spark that grew into the ideas reported here.

References

Hyunah Baek. 2018. Computational representation of unbounded stress: Tiers with structural features. In *Proceedings of CLS 53*, pages 13–24.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49:23–60.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Formal Grammar*, pages 35–50, Berlin, Heidelberg. Springer.

Saul Gorn. 1967. Explicit definitions and linguistic dominoes. In *Systems and Computer Science, Proceedings of the Conference held at University of Western Ontario, 1965*, Toronto. University of Toronto Press.

Thomas Graf. 2011. Closure properties of Minimalist derivation tree languages. In *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 96–111, Heidelberg. Springer.

Thomas Graf. 2017. A computational guide to the dichotomy of features and constraints. *Glossa*, 2:1–36.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, pages 117–136.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of SIGMORPHON 2018*, pages 151–160.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

Morris Halle and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Ken Hale and Samuel J. Keyser, editors, *The view from building 20*, pages 111–176. MIT Press, Cambridge, MA.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Gregory M. Kobele. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 129–144.

Marcus Kracht. 1997. Inessential features. In Alain Lecomte, F. Lamarche, and G. Perrier, editors, *Logical Aspects of Computational Linguistics*, pages 43–62. Springer, Berlin.

Lei Liu. 2018. Minimalist parsing of heavy NP shift. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*. Association for Computational Linguistics.

James Rogers. 1997. Strict LT_2 : Regular :: Local : Recognizable. In *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 (Selected Papers)*, volume 1328 of *Lectures Notes in Computer Science/Lectures Notes in Artificial Intelligence*, pages 366–385. Springer.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Hans-Jörg Tiede. 2008. Inessential features, ineliminable features, and modal logics for model theoretic syntax. *Journal of Logic, Language and Information*, 17:217–227.

John Torr. 2017. Autobank: a semi-automatic annotation tool for developing deep Minimalist grammar treebanks. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–86.

Mai Ha Vu. 2018. Towards a formal description of NPI-licensing patterns. In *Proceedings of the Society for Computation in Linguistics*, volume 1, pages 154–163.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.

Modeling the learning of the Person Case Constraint

Adam Liter
Dept. of Linguistics
University of Maryland
College Park, MD 20742 USA
io@adamliter.org

Naomi H. Feldman
Dept. of Linguistics & UMIACS
University of Maryland
College Park, MD 20742 USA
nhf@umd.edu

Abstract

Many domains of linguistic research posit feature bundles as an explanation for various phenomena. Such hypotheses are often evaluated on their simplicity (or parsimony). We take a complementary approach. Specifically, we evaluate different hypotheses about the representation of person features in syntax on the basis of their implications for learning the Person Case Constraint (PCC). The PCC refers to a phenomenon where certain combinations of clitics (pronominal bound morphemes) are disallowed with ditransitive verbs. We compare a simple theory of the PCC, where person features are represented as atomic units, to a feature-based theory of the PCC, where person features are represented as feature bundles. We use Bayesian modeling to compare these theories, using data based on realistic proportions of clitic combinations from child-directed speech. We find that both theories can learn the target grammar given enough data, but that the feature-based theory requires significantly less data, suggesting that developmental trajectories could provide insight into syntactic representations in this domain.

1 Introduction

Representing surface realizations as bundles of features is ubiquitous in linguistics. For example, in syntax, different forms that result from subject-verb agreement are taken to be the result of different feature bundles. Relevant features for subject verb agreement in English include at least the tense and the number of the subject. Although there is little variation in the different surface forms for English verbs, the verb *walk* does differ when the subject is singular and the tense is present (*walks*), compared to when the subject is singular and the tense is past (*walked*).

Features are often taken to be either privative or binary (though these are not the only possibilities). For example, some might argue that the English singular/plural distinction is based on a privative feature: a noun phrase can either be specified as plural or not specified for number (*e.g.*, [plural] and [ ]). In this case, when *dog* is marked with "[plural]", it is realized as *dogs*. Others might argue that the distinction is based on a binary feature: a noun phrase can be specified as "plus" or "minus" (*e.g.*, [+plural] and [−plural]). In this case, when *dog* is marked with "[+plural]", it is realized as *dogs*.

Feature representations are typically evaluated based on the extent to which they simplify linguistic analyses, that is, on their ability to provide parsimonious descriptions of cross-linguistic grammatical patterns. For a concrete example of this type of argument, see Adger and Smith (2010), who argue that both the intra-dialectal variation in the inflection of the verb *be* in Buckie Scottish English as well as the inter-dialectcal variation in the inflection of the verb *be* in English more broadly is nicely explained by a feature system involving binary-valued features of Singular, Participant, and Author.

In this paper, we take a different approach to evaluating feature representations, focusing on their implications for learning (for similar approaches, see Pearl and Sprouse, 2013; Pearl et al., 2017; Rasin and Katzir, 2017; Pearl and Sprouse, 2019). Specifically, we investigate how person features might be represented in the syntactic component of the grammar, using the domain of clitics as a case study and the learnability of a phenomenon involving clitics as a metric for plausibility. We find that both of the representational theories that we test can learn the target grammar given enough data, but that they differ considerably in the amount of data they require. This suggests that children's learning trajectory has the potential to provide insight into syntactic representations in this domain.

2 The Person Case Constraint

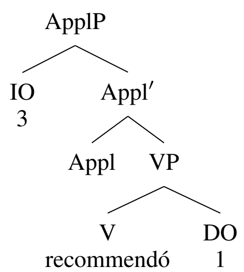
Clitics are bound morphemes (*i.e.*, morphemes that cannot stand on their own). The clitics relevant to the PCC are pronominal clitics, which encode first, second, and third person, and they must occur immediately next to a free morpheme, usually a verb. For example, (1) shows a Spanish sentence where the direct and indirect objects of a ditransitive verb are both realized as clitics. The clitics are immediately before the verb, and, in this case, they encode first and third person, respectively.

(1) Me lo cuentas
1.SG.DAT 3.SG.ACC tell
'(You) tell it to me'

Interestingly, when the direct and indirect objects to a ditransitive verb are both realized as clitics, not all combinations are possible. For example, compare (1) to (2), where the first person clitic serves as the direct object and the third person clitic serves as the indirect object (*i.e.*, the opposite of (1)). The sentence in (2) is ungrammatical.[1]

(2) * Me le recommendó
1.SG.ACC 3.SG.DAT recommend.PST
'S/he recommended me to her/him'

[1]Note that even though the first person clitic occurs before the third person clitic in both (1) and (2), the literature usually talks about the ungrammaticality of (2) with the starred string "*3 1". This is meant to indicate the underlying argument structure relations—namely, in the syntactic analysis of (2), but not (1), the dative third person argument is structurally higher than first person argument, as shown in (i). This is generally written as "*3 1", meant to reflect the fact that the third person argument structurally precedes the first person argument, even though the surface string order of the clitics is the opposite.

(i) *


Nonetheless, as (1) shows, there are some instances where the surface string order of the clitics does match the underlying argument structure relations. This depends on a variety of language specific factors, including at least the nature of the particular verb and ordering effects between some of the clitics in some languages.

The ungrammaticality of (2) is part of a broader phenomenon called the Person Case Constraint (PCC) (see, *e.g.*, Bonet, 1991, 1994); the PCC will be the central focus of our case study on the representation of person features.

Ignoring the possible combinations of direct and indirect objects with either both first person or both second person arguments[2] gives seven different possible direct and indirect object pairings: 1 2, 1 3, 2 1, 2 3, 3 1, 3 2, and 3 3. There are four attested variants of the PCC, each of them banning a different subset of these seven possible clitic combinations. The four variants of the PCC (and their names) are given in Table 1, along with languages/dialects that are known to instantiate each of them (note that these tables include 3 3 and thus differ slightly from those reported in Graf, 2012, p. 86).

Because there are different variants of the PCC that occur cross-linguistically, a child will have to learn which variant their language instantiates on the basis of input.

3 Evaluating two theories of the PCC

We use a Bayesian learning model to evaluate the plausibility of two theories of the representation of person features. The first theory is one in which first, second, and third person have no further structure; they are just represented as atomic features in the grammar, like in (3). We refer to this as the simple theory of the PCC because the grammar is assumed to simply state, for each possible clitic combination, whether it is grammatical.

(3) a. $1 = 1$
b. $2 = 2$
c. $3 = 3$

We compare this to another theory in which first, second, and third person are represented as feature bundles, consisting of two values, one for the binary feature Author and one for the binary feature Participant, as in (4) (Nevins, 2007). We refer to this as the feature-based theory of the PCC.

(4) a. $1 = \begin{bmatrix} +\text{Auth} \\ +\text{Part} \end{bmatrix}$
b. $2 = \begin{bmatrix} -\text{Auth} \\ +\text{Part} \end{bmatrix}$

[2]The combinations with both first or both second person arguments are often ignored in this literature because of other complicating factors. Specifically, these combinations are also governed by another part of the grammar, Binding Theory (see, *e.g.*, Chomsky, 1981).

c. $3 = \begin{bmatrix} -\text{Auth} \\ -\text{Part} \end{bmatrix}$

Based on corpus data from child-directed speech, we model the learning of one PCC variant in order to investigate the plausibility of these different representations of person features. The remainder of this section lays out these two representational theories in more detail.

3.1 A simple theory of the PCC

The simple theory of the PCC states, for each clitic combination, whether or not it is grammatical. For this theory, person features are atomic (cf. (3)), and the grammar simply states that some combinations (*e.g.*, *2 1) are banned. Given that there are 7 clitic combinations, this leads to $2^7 = 128$ possible grammars, some of which are shown in Table 2.[3]

Grammar	1 2	1 3	2 1	2 3	3 1	3 2	3 3
SG_1	✓	✓	✓	✓	✓	✓	✓
SG_2	✓	✓	✓	✓	✓	✓	*
SG_3	✓	✓	✓	✓	✓	*	✓
SG_4	✓	✓	✓	✓	✓	*	*
SG_5	✓	✓	✓	✓	*	✓	✓
SG_6	✓	✓	✓	✓	*	✓	*
SG_7	✓	✓	✓	✓	*	*	✓
SG_8	✓	✓	✓	✓	*	*	*
...	...	...	...	...	...	...	...
SG_{21}	✓	✓	*	✓	*	✓	✓
SG_{22}	✓	✓	*	✓	*	✓	*
SG_{23}	✓	✓	*	✓	*	*	✓
...	...	...	...	...	...	...	...
SG_{55}	*	✓	*	✓	*	*	✓
...	...	...	...	...	...	...	...
SG_{85}	*	✓	*	✓	*	✓	✓
SG_{86}	*	✓	*	✓	*	✓	*
SG_{87}	*	✓	*	✓	*	*	✓
...	...	...	...	...	...	...	...
SG_{128}	*	*	*	*	*	*	*

Table 2: Some of the 128 possible simple grammars (SG) for the PCC

3.2 A feature-based theory of the PCC

Nevins (2007) proposes a feature-based theory of the four PCC varieties. This theory is much more

IO↓/DO→	1	2	3
1	NA	*	✓
2	*	NA	✓
3	*	*	✓

(a) Strong PCC (Greek, Spanish, *etc.*)

IO↓/DO→	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	*	✓

(b) Ultrastrong PCC (Classical Arabic, Spanish, *etc.*)

IO↓/DO→	1	2	3
1	NA	✓	✓
2	✓	NA	✓
3	*	*	✓

(c) Weak PCC (French, Catalan, Spanish, *etc.*)

IO↓/DO→	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	✓	✓

(d) Me-First PCC (Romanian, Spanish, *etc.*)

Table 1: PCC varieties (rows indicate the indirect object, and columns indicate the direct object; '✓' indicates grammatical, and '*' indicates ungrammatical; for example, *1 2 is ungrammatical in Strong PCC languages but grammatical in all other PCC varieties)

[3]The simple grammar for the Strong PCC would be SG_{55}, the simple grammar for the Ultrastrong PCC would be SG_{23}, the simple grammar for the Weak PCC would be SG_7, and the simple grammar for the Me-First PCC would be SG_{21}.

restrictive in that it allows many fewer possible types of grammars. For this theory, it is crucial that first, second, and third person are represented as feature bundles, consisting of two binary feature values, as shown above in (4).

The features Author and Participant are taken to be primitive features in the theory of morphosyntax, and each can be valued as either + or −.[4] Broadly, this theory relies on how these features bundles can (or cannot) co-occur with one another in concert with a syntactic operation called Agree.

To spell out the details more carefully, clitics are understood to be the morphophonological realization of a syntactic operation called Agree (see, *e.g.*, Borer, 1984). The possible grammars in this feature-based theory thus consist of different possible specifications for the feature(s) that trigger(s) Agree. Specifically, there is a syntactic probe, *v*, that, when introduced into the derivation, triggers Agree. Nevins assumes that the probe can be specified to search for either marked and/or contrastive Author and Participant features (cf. Calabrese, 1995; Nevins, 2007, p. 285–290).

The marked version of each feature is its + value. A contrastive instance of the Participant feature is one that occurs in the presence of −Auth; when Participant occurs with +Auth, it is not contrastive because there is no possible feature bundle $\begin{bmatrix} +\text{Auth} \\ -\text{Part} \end{bmatrix}$ (cf. fn. 4). In other words, if the feature bundle contains +Auth, it must necessarily also contain +Part. A contrastive instance of the Author feature is one that occurs in the presence of +Part; *i.e.*, when you have a feature bundle that contains −Part, then it must necessarily also contain −Auth.

Given this theory of clitics and the PCC, there are then nine possible feature-based grammars (FG), which are all given the first column of Table 3. In the grammar specifications in this table, 'u' indicates that the probe, *v*, is looking for a feature of the type that follows the 'u' to Agree with.[5] Furthermore, we indicate, for example, contrastive Author as 'uAuth/[+Part]', which can be read as "the probe is looking for an Author feature that occurs in the context of +Part".

Here, we walk through two example derivations. For further discussion and derivations, see Nevins (2007, p. 290–301).

Let's first consider the clitic order *1 2, which is disallowed in Strong PCC languages. The feature specification that is claimed to give rise to Strong PCC languages is FG_6.

Nevins argues that there are two conditions that govern the application of Agree (2007, p. 295), Contiguous Agree and Matched Values.

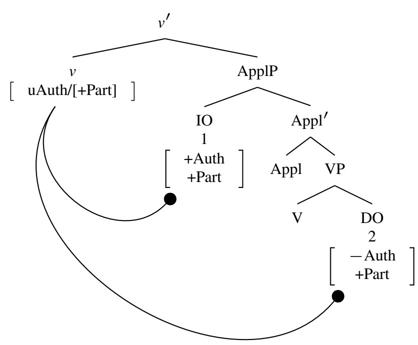
(5) Contiguous Agree: For a relativization R of a feature F on a Probe P, and $x \in$ Domain(R(F)), $\neg\exists y$, such that $y > x$ and $p > y$ and $y \in$ Domain(R(F))
"There can be no interveners between P and x that are not in the domain of relativization that includes x"

(6) Matched Values: For a relativization R of a feature F, $\exists\alpha$, $\alpha \in \{+, -\}$, $\forall x$, $x \in$ Domain(R(F)), val(x,F)$= \alpha$
"All elements within the domain of relativization must contain the same value"

In other words, Contiguous Agree requires that any argument that occurs in between the probe and the target of Agree must also itself be a target of Agree, and Matched Values requires that all arguments that are in the domain of the Agree operation must share the same value (*e.g.*, both must be +Auth; one cannot be −Auth and the other +Auth).

Now, in the case of *1 2 when the grammar is FG_6 (*i.e.*, the Strong PCC), where the probe, *v*, seeks to Agree with arguments bearing contrastive Author, a partial derivation will look like the one in (7).

(7) *



In this case, the condition Matched Values is violated. Both the first person indirect object and the second person direct object are in the domain of

[4]The feature combination of $\begin{bmatrix} +\text{Auth} \\ -\text{Part} \end{bmatrix}$ is taken to be impossible because of what the features mean—namely, it's not possible to be the author (*i.e.*, speaker) in a conversation but not a participant in that same conversation.

[5]This is generally understood to mean "uninterpretable" in the syntactic literature; for an overview of feature theory in Minimalist theories of syntax, see Pesetsky and Torrego (2007).

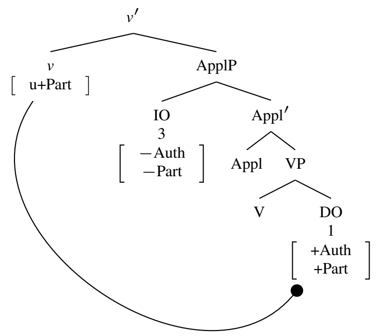
Probe	Grammar	1 2	1 3	2 1	2 3	3 1	3 2	3 3
$v\begin{bmatrix} \ \end{bmatrix}$	FG_1	✓	✓	✓	✓	✓	✓	✓
$v\begin{bmatrix} \text{u+Part} \end{bmatrix}$	FG_2	✓	✓	✓	✓	*	*	✓
$v\begin{bmatrix} \text{u+Auth} \end{bmatrix}$	FG_3	✓	✓	*	✓	*	✓	✓
$v\begin{bmatrix} \text{u+Part} \\ \text{u+Auth} \end{bmatrix}$	FG_4	✓	✓	*	✓	*	*	✓
$v\begin{bmatrix} \text{uAuth/[+Part]} \\ \text{uPart/[}-\text{Auth]} \end{bmatrix}$	FG_5	*	*	*	*	*	*	✓
$v\begin{bmatrix} \text{uAuth/[+Part]} \end{bmatrix}$	FG_6	*	✓	*	✓	*	*	✓
$v\begin{bmatrix} \text{uAuth/[+Part]} \\ \text{u+Part} \end{bmatrix}$	FG_7	*	✓	*	✓	*	*	✓
$v\begin{bmatrix} \text{uPart/[}-\text{Auth]} \end{bmatrix}$	FG_8	*	*	✓	*	✓	*	✓
$v\begin{bmatrix} \text{uPart/[}-\text{Auth]} \\ \text{u+Auth} \end{bmatrix}$	FG_9	*	*	*	*	*	*	✓

Table 3: The 9 possible feature-based (FG) grammars for the PCC, according to Nevins (2007)

Agree for the feature uAuth/[+Part] on the probe, v (because they both have Author features that occur in the context of +Part). However, they have differing values for Author, so Matched Values is violated, giving rise to the ungrammaticality of *1 2 when the grammar is FG_6.

Next, let's consider the case of *3 1, which is disallowed in Weak PCC languages. The feature specification that is claimed to give rise to Weak PCC languages is FG_2, where the probe, v, seeks to Agree with arguments bearing a marked Participant feature. In the case of *3 1 when the grammar is FG_2, a partial derivation will look like the one in (8).

(8) *



Here, the probe is looking for a +Part feature; this means that it can agree with the direct object; however, there is a structurally higher element—namely, the third person indirect object, $\begin{bmatrix} -\text{Auth} \\ -\text{Part} \end{bmatrix}$—that intervenes between the probe, v, and the target of Agree but is not in the domain of the probe because it does not contain a +Part feature. This violates the condition Contiguous Agree, so the clitic order *3 1 is thereby disallowed in FG_2.

Walking through the derivations for all seven possible clitic orders for all nine feature-based grammars gives the results shown in Table 3.[6]

4 The learning model

We use Bayesian modeling to implement a computational-level learning model that infers a grammar, given a bunch of sentences with ditransitive verbs and two clitics. In the case of the feature-based theory of the PCC, there are 9 grammars, and so the hypothesis space is much smaller. In the case of the simple theory of the PCC, there are 128 grammars, and so the hypothesis space is much larger.

Using realistic proportions of the occurrences of these types of constructions in child-directed speech, we seek to establish how much data would be needed to learn the correct grammar under each of these theories.

[6]The feature-based grammar for the Strong PCC would be FG_6, as noted, or FG_7 (these two feature-based grammars are extensionally equivalent), the feature-based grammar for the Ultrastrong PCC would be FG_4, the feature-based grammar for the Weak PCC would be FG_2, and the feature-based grammar for the Me-First PCC would be FG_3. The remaining grammars would then delimit the predicted typology of PCC languages. FG_1 would be a language without PCC effects (and perhaps also without clitics), like English; there would be two further predicted types of PCC languages, FG_8, which Nevins calls a "Me-Last" language, and FG_5 and FG_9, which are extensionally equivalent in only allowing 3 3 (note that Nevins (2007) does not consider 3 3 constructions).

4.1 The generative model

We assume the generative model depicted in Figure 1. A generative model encodes the assumptions a learner would have about how the data it observes are generated.

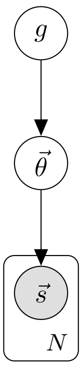


Figure 1: Generative model

Our generative model assumes that there is a grammar, g, that determines how often certain clitic combinations will be used. In the case of the simple theory of the PCC, g will be one of $SG_1, \ldots, SG_{128}$, and in the case of the feature-based theory of the PCC, g will be one of $FG_1, \ldots, FG_9$.

This grammar g is assumed to generate a vector of probabilities, $\vec{\theta}$, which governs the frequency of use of each of the different clitic combinations in the language. In other words, $\vec{\theta}$ determines how often one would expect to see each clitic combination in a corpus containing N ditransitive sentences that have cliticized both internal arguments. In our model, we assume that the elements of $\vec{\theta}$ corresponding to any clitic orderings that are disallowed under g are set to zero, and that the remaining elements of $\vec{\theta}$ are generated from a Dirichlet distribution with dimensionality equal to the number of permitted clitic orderings,

$$\vec{\theta} \mid g \sim \text{Dir}(\langle 1, \ldots, 1 \rangle) \qquad (1)$$

This Dirichlet distribution encodes a belief that any value of $\vec{\theta}$ that is consistent with the grammar is equally likely, a priori.

The instances of clitic combinations that a learner observes, represented in our generative model as $\vec{s}$, are then assumed to be sampled from $\vec{\theta}$. For example, if, in the corpus, there were 3 instances of the 1 3 clitic combination, 6 instances of the 3 3 clitic combination, and no others, then $\vec{s}$ would be $\langle 0, 3, 0, 0, 0, 0, 6 \rangle$. The generative model assumes that $\vec{s}$ are sampled from a multinomial distribution with parameter $\vec{\theta}$,

$$\vec{s} \mid \vec{\theta} \sim \text{Multinom}(N, \vec{\theta}) \qquad (2)$$

The learner observes the clitic combinations in its corpus and infers which of the possible grammars was most likely to have generated these data.

4.2 Inferring the grammar

Given a count of the occurrence of each of the seven possible clitic orders, $\vec{s}$, from a corpus of sentences, the posterior probability of each possible grammar, $p(g \mid \vec{s})$, can be computed. Using Bayes' rule, $p(g \mid \vec{s})$ can be calculated as

$$p(g \mid \vec{s}) = \frac{p(\vec{s} \mid g)p(g)}{\sum_{g'} p(\vec{s} \mid g')p(g')} \qquad (3)$$

We assume a uniform prior probability distribution over grammars, $p(g)$. The likelihood term, $p(\vec{s} \mid g)$, is calculated by integrating over all possible values of $\vec{\theta}$,

$$p(\vec{s} \mid g) = \int p(\vec{s} \mid \vec{\theta})p(\vec{\theta} \mid g)\mathrm{d}\vec{\theta} \qquad (4)$$

Note that the complexity of each hypothesized grammar differs because in grammars that rule out some clitic combinations, the corresponding values of $\vec{\theta}$ are set to zero, and the corresponding likelihood terms have fewer values of θ to integrate over. Because of this, a grammar that allows fewer clitic combinations will have a higher likelihood than a grammar that allows more clitic combinations, when some counts in $\vec{s}$ are zero (cf. Tenenbaum and Griffiths, 2001). This is so because a more complex grammar needs to integrate over values of $\vec{\theta}$ that give probability to things that do not occur in the learner's input.

For example, in trying to determine how likely it is that g is either SG_1 or FG_1, both which allow all 7 possible clitic combinations, $p(\vec{s} \mid \vec{\theta})$ is $\frac{N!}{n_1! \cdots n_7!} \prod_{i=1}^{7} \theta_i^{n_i}$, and $p(\vec{\theta} \mid g)$ is $\frac{\Gamma(\sum_{i=1}^{7} \alpha_i)}{\prod_{i=1}^{7} \Gamma(\alpha_i)} \prod_{i=1}^{7} \theta_i^{\alpha_i - 1}$. On the other hand, if trying to determine how likely it is that g is either FG_3 or SG_{21}, both which allow 5 of the 7 possible clitic combinations, $p(\vec{s} \mid \vec{\theta})$ will be $\frac{N!}{n_1! \cdots n_5!} \prod_{i=1}^{5} \theta_i^{n_i}$, and $p(\vec{\theta} \mid g)$ will be $\frac{\Gamma(\sum_{i=1}^{5} \alpha_i)}{\prod_{i=1}^{5} \Gamma(\alpha_i)} \prod_{i=1}^{5} \theta_i^{\alpha_i - 1}$.

To calculate the likelihood that g is, for example, FG_1, we can substitute these terms into Eq. 4, which yields Eq. 5 (cf. Gelman et al., 2014).

$$\frac{\prod_{i=1}^{7}\Gamma(n_i+\alpha_i)}{\Gamma\left(\sum_{i=1}^{7} n_i+\alpha_i\right)}\frac{N!}{n_1!\cdots n_7!}\frac{\Gamma\left(\sum_{i=1}^{7}\alpha_i\right)}{\prod_{i=1}^{7}\Gamma(\alpha_i)} \quad (5)$$

On the other hand, if calculating the likelihood that g is instead FG_3, then all of the instances of '7' in Eq. 5 would be replaced with '5'.

Having defined the learning model, we can now give it data to learn from, based on child-directed speech, and see what difference the size of the hypothesis space makes.

5 Simulations

We conducted several simulations based on realistic proportions of clitic combinations taken from child-directed speech.

5.1 Data

We estimated the frequency of each clitic combination in child-directed speech based on their distribution in the Aguirre Corpus (Aguirre, 2003), from CHILDES (MacWhinney, 2000). This corpus contains 30 files for one Spanish-speaking child between the ages of 1;7 and 2;10. We extracted the 13,411 child-directed utterances from the files using the Python package `PyLangAcq` (Lee et al., 2016). Then, we used the Python package `spaCy` (Honnibal and Montani, 2017) to parse these utterances. This allowed us to extract utterances where two clitics preceded a verb; *i.e.*, we extracted the sentences with clitic clusters that are relevant for learning the PCC. We found 50 instances of 1 3, 148 instances of 2 3, 4 instances of 3 2, and 68 instances of 3 3. This indicates that the speakers in this corpus speak a Me-First PCC language, since these constructions are only compatible with that kind of PCC language. We failed to observe any instances of 1 2, even though this construction is grammatical in Me-First PCC languages (cf. Table 1).

Training corpora for our models were created based on the frequency distribution found in the Aguirre Corpus. Because counts from this corpus were used as the weights for the random sampling, we applied smoothing so that the simulations had some probability of including the 1 2 construction, which is grammatical in Me-First PCC languages (again, cf. Table 1) but had a zero count in the Aguirre corpus. The smoothing consisted of adding 0.1 to all of the counts for grammatical constructions from the Aguirre corpus. For each simulation, we randomly sampled n PCC constructions with weights based on the smoothed frequency profile found in the Aguirre corpus; we did this for three values of n: 66, 666, and 6,666. These values were chosen because Hart and Risley (1995) estimate that children hear 333,333 utterances per year in their first three years of life. Moreover, 2% of the utterances in the Aguirre Corpus were relevant for learning the PCC, so 2% of 333,333 is 6,666 (see subsection 5.3 for more discussion).

5.2 Results

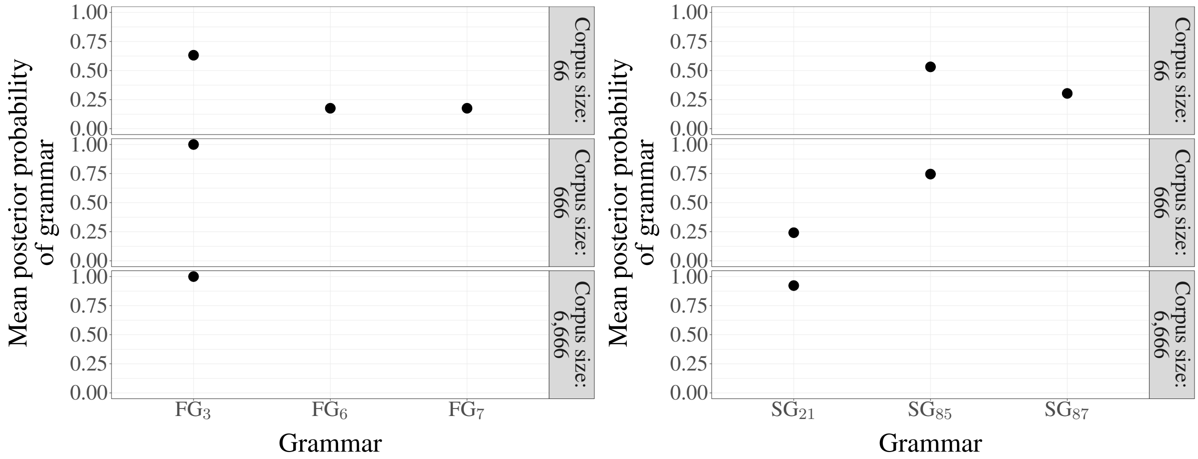
We trained Simple learning models and Feature-based learning models. Each model used the data that we generated on the basis of the Aguirre corpus to compute a posterior distribution over all the grammars in its hypothesis space. We ran 1,000 replications of each model at each corpus size, n, and we averaged the results of these 1,000 replications. These mean posterior probabilities are plotted in Figure 2 (to make the plots more readable, only grammars with a posterior probability equal to or greater than 0.1 are plotted).

As can be seen in Figure 2, the grammar with the highest posterior probability is the correct grammar for all three corpus sizes under the Feature-based learning model. That is to say, in these cases, the model has converged on FG_3, which is the feature-based grammar for the Me-First PCC (cf. fn. 6).

On the other hand, for the Simple learning model, the simulations converge on SG_{85} when the corpus size is 66 and 666, but the simple grammar that instantiates the Me-First PCC variety is in fact SG_{21}; SG_{85} differs from SG_{21} in disallowing 1 2. (SG_{87} furthermore disallows 3 2, compared to SG_{21}; see Table 2.) Nonetheless, when the corpus size is 6,666, the Simple learning model does correctly converge on SG_{21}.

5.3 Discussion

In our simulation results, we saw that the Feature-based learning model is able to converge on the correct grammar much quicker than the Simple learning model. In fact, if data are sparse, the Simple learning model converges on unattested PCC varieties. The Simple learning model clearly needs more data to learn the target grammar.



(a) Feature-based learning model results

(b) Simple learning model results

Figure 2: Mean posterior probabilities for learning simulations (FG_3 is the target grammar for the feature-based theory; SG_{21} is the target grammar for the simple theory)

As noted, we chose the corpus sizes that we did because Hart and Risley (1995) estimate that children hear 1 million utterances in their first 3 years, or 333,333 utterances per year.[7] Moreover, the Aguirre corpus contained 13,411 child-directed utterances, and we found 270 utterances with clitic clusters, which is $\approx 2\%$. Two percent of 333,333 is 6,666. Thus, a young learner might hear 6,666 clitic combinations in one of their early years of life.

This suggests that the Simple learning model may in fact have enough data that it needs in order to converge on the correct target grammar, but there are several things one would want to further investigate. First, one would want to know when a child has fully acquired the PCC restrictions of their language. To the best of our knowledge, there is very little research on this. Tsakali and Wexler (2010) reported that Greek-acquiring children seem to know the PCC restrictions of their language by age 5, but they tested this by eliciting acceptability judgments, which are often hard to do with younger children. At best, this might be an upper bound for when children know the PCC restrictions of their language. Indeed, Blasco (2000) showed that Spanish-acquiring children were correctly producing both accusative and dative clitics in Spanish by the age of 2;2, if not earlier.[8] Whether this means that they know the PCC restrictions at such a young age is an open question.

Second, there is a difference between input and intake (cf. Omaki and Lidz, 2015); that is to say, just because a learner hears 6,666 clitic clusters, does not mean that the learner uses those utterances for learning. A learner might be inattentive, a learner might fail to perceive a given utterance, a learner might fail to parse a given utterance, *etc.*. Especially at a very early age, when the child hasn't yet learned the syllable structure of their language and how to identify morpheme boundaries, it seems unlikely that the child would learn anything about the PCC variant of their target language upon hearing a clitic cluster in their input.

Moreover, as can be seen by examples (1) and (2), the surface string order does not necessarily reflect the underlying argument structure relations, which can interact with other language specific factors in a variety of ways. For example, in many dialects of Spanish, the clitics must occur in a certain order, regardless of the underlying argument structure relations (cf. fn. 1). Absent definitive knowledge of both the argument structure of the verb and such language specific factors as clitic ordering effects, it might be advantageous for a learner to ignore some of its input (cf. Perkins et al., 2017).

Thus, if a child really did know the PCC variant of their target language by age 2;2, our results might argue against the Simple learning model, if not all of the clitic clusters in the child's input are taken up and used for learning. Nevertheless, there is much we don't yet know about the acquisition of

[7] These estimates are for American children who are acquiring English, but presumably the order of magnitude is comparable for learners of other languages, such as Spanish.

[8] For further discussion on the acquisition of clitics more generally, see Tsakali (2014).

the PCC.

Additionally, there is more that could be done on the modeling side of things. For example, the models we've presented abstract away from additional complexities of the assumed grammars, such as the necessity of the Agree operation for Nevins's (2007) theory of the PCC or the necessity of the features Author and Participant. If such additional complexities also need to be learned, (*i.e.*, if they are not already known at the time when PCC learning begins), one would want to create learning models that include these complexities and run further simulations.

Ultimately, this work is intended as a computational-level analysis that begins to help set an upper bound on how much data children would need to use in order to learn the PCC, given particular theoretical and representational assumptions. We've compared the feature representations assumed by Nevins's (2007) feature-based theory to the feature representations assumed in a simple theory of the PCC. In addition to Nevins's (2007) theory, there are other more restrictive theories of the PCC (*e.g.*, Béjar and Rezac, 2003; Pancheva and Zubizarreta, 2018; Graf, 2019); so future modeling work should also seek to establish upper bounds for the theoretical and representational assumptions of these analyses. Given that they're more restrictive theories, one might expect the results to be similar to the results for the Feature-based learning models reported here, but such modeling work may nevertheless help distinguish between them, when coupled with better information about the acquisition of the PCC.

6 Conclusion

In this paper, we used a learning model to investigate how person features might be represented in the syntactic component of the grammar. We compared two possibilities: one where the person features are represented as atomic units (cf. (3)) and one where the person features are represented as feature bundles, consisting of values for the binary features Author and Participant (cf. (3)).

We simulated different-sized corpora based on realistic distributions in the input to children and evaluated these learning models against the simulated data. We found that the Feature-based learning model is able to learn the target grammar much quicker than the Simple learning model. Given enough data, the Simple learning model will converge on the correct grammar; however, if data are sparse, the Simple learning model will converge on unattested PCC variants, which might tell against the simple theory of the PCC. That is, this suggests that the larger hypothesis space, in addition to being possibly unparsimonious, may lead learners astray, particularly if data are sparse.

One would particularly want to know how much input the child actually gets, how much of that the child uses, and when the child has fully acquired the PCC restrictions. Such information, coupled with our results, would inform whether one of these ideas about the representation of person features in the grammar is more plausible than another.

Acknowledgments

We gratefully acknowledge the support of NSF NRT award DGE-1449815, which has supported the first author.

References

David Adger and Jennifer Smith. 2010. Variation in agreement: A lexical feature-based approach. *Lingua*, 120(5):1109–1134.

Carmen Aguirre. 2003. Early verb development in one Spanish-speaking child. In Dagmar Bittner, Wolfgang U. Dressler, and Marianne Kilani-Schoch, editors, *Development of Verb Inflection in First Language Acquisition: A Cross-Linguistic Perspective*, number 21 in Studies on Language Acquisition, pages 1–26. Mouton de Gruyter, Berlin, Germany.

Susana Béjar and Milan Rezac. 2003. Person licensing and the derivation of PCC effects. In Ana Teresa Pérez-Leroux and Yves Roberge, editors, *Romance Linguistics: Theory and Acquisition*, number 244 in Current Issues in Linguistic Theory, pages 49–62. John Benjamins Publishing Company, Amsterdam, The Netherlands.

Maria Blasco. 2000. *The Acquisition of Pronominal Object Clitics in Spanish*. Ph.D. thesis, The City University of New York, New York, NY.

M. Eulàlia Bonet. 1991. *Morphology after Syntax: Pronominal Clitics in Romance*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

M. Eulàlia Bonet. 1994. The person-case constraint: A morphological approach. In Heidi Harley and Colin Phillips, editors, *The Morphology-Syntax Connection*, number 22 in MIT Working Papers in Linguistics, pages 33–52. Cambridge, MA.

Hagit Borer. 1984. *Parametric Syntax: Case Studies in Semitic and Romance Languages*. Number 13 in Studies in Generative Grammar. Foris Publications, Dordrecht, The Netherlands.

Andrea Calabrese. 1995. A constraint-based theory of phonological markedness and simplification procedures. *Linguistic Inquiry*, 26(3):373–463.

Noam Chomsky. 1981. *Lectures on Government and Binding*. Foris, Dordrecht, The Netherlands.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2014. *Bayesian Data Analysis*, 3rd edition. Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL.

Thomas Graf. 2012. An algebraic perspective on the person case constraint. In Thomas Graf, Denis Paperno, Anna Szabolcsi, and Jos Tellings, editors, *Theories of Everything: In Honor of Ed Keenan*, number 17 in UCLA Working Papers in Linguistics, pages 85–90.

Thomas Graf. 2019. Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling*, 7(2):3–47.

Betty Hart and Todd R. Risley. 1995. *Meaningful Differences in the Everyday Experience of Young American Children*. Paul H. Brookes Publishing Co., Inc., Baltimore, MD.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jackson L. Lee, Ross Burkholder, Gallagher B. Flinn, and Emily R. Coppess. 2016. Working with chat transcripts in python. Technical Report TR-2016-02, Department of Computer Science, University of Chicago.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, 3rd edition. Lawrence Erlbaum Associates, Mahwah, NJ.

Andrew Nevins. 2007. The representation of third person and its consequences for person-case effects. *Natural Language & Linguistic Theory*, 25(2):273–313.

Akira Omaki and Jeffrey Lidz. 2015. Linking parser development to acquisition of syntactic knowledge. *Language Acquisition*, 22(2):158–192.

Roumyana Pancheva and Maria Luisa Zubizarreta. 2018. The person case constraint: The syntactic encoding of perspective. *Natural Language & Linguistic Theory*, 36(4):1291–1337.

Lisa Pearl, Timothy Ho, and Zephyr Detrano. 2017. An argument from acquisition: Comparing English metrical stress representations by how learnable they are from child-directed speech. *Language Acquisition*, 24(4):307–342.

Lisa Pearl and Jon Sprouse. 2013. Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, 20(1):23–68.

Lisa Pearl and Jon Sprouse. 2019. The acquisition of linking theories: A tolerance principle approach to learning UTAH and rUTATH. Ms., University of California, Irvine, CA and University of Connecticut, Storrs, CT.

Laurel Perkins, Naomi H. Feldman, and Jeffrey Lidz. 2017. Learning an input filter for argument structure acquisition. In *Proceedings of the 7th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2017)*, pages 11–19.

David Pesetsky and Esther Torrego. 2007. The syntax of valuation and the interpretability of features. In Simin Karimi, Vida Samiian, and Wendy K. Wilkins, editors, *Phrasal and Clausal Architecture: Syntactic Derivation and Interpretation*, volume 101 of *Linguistik Aktuell/Linguistics Today*, pages 262–294. John Benjamins Publishing Company, Amsterdam, The Netherlands.

Ezer Rasin and Roni Katzir. 2017. A learnability argument for constraints on underlying representations. Ms., Leipzig University and Tel Aviv University. Available at `https://ling.auf.net/lingbuzz/002260`.

Joshua B. Tenenbaum and Thomas L. Griffiths. 2001. Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, 24(4):629–640.

Vina Tsakali. 2014. Acquisition of clitics: The state of the art. In Kleanthes K. Grohmann and Theoni Neokleous, editors, *Developments in the Acquisition of Clitics*, chapter 5, pages 161–187. Cambridge Scholars Publishing, Newcastle, UK.

Vina Tsakali and Kenneth Wexler. 2010. The acquisition of Person Case Constraint in Greek. Paper presented at the *19th International Symposium on Theoretical and Applied Linguistics*, Thessaloniki, Greece.

A Closer Look at the Performance of Neural Language Models on Reflexive Anaphor Licensing

Jennifer Hu
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA
jennhu@mit.edu

Sherry Yong Chen
Dept. of Linguistics and Philosophy
Massachusetts Institute of Technology
Cambridge, MA
sychen@mit.edu

Roger Levy
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA
rplevy@mit.edu

Abstract

An emerging line of work uses psycholinguistic methods to evaluate the syntactic generalizations acquired by neural language models (NLMs). While this approach has shown NLMs to be capable of learning a wide range of linguistic knowledge, confounds in the design of previous experiments may have obscured the potential of NLMs to learn certain grammatical phenomena. Here we re-evaluate the performance of a range of NLMs on reflexive anaphor licensing. Under our paradigm, the models consistently show stronger evidence of learning than reported in previous work. Our approach demonstrates the value of well-controlled psycholinguistic methods in gaining a fine-grained understanding of NLM learning potential.[1]

1 Introduction

To gain a deeper understanding of the grammatical generalizations acquired by neural language models (NLMs), an emerging line of work seeks to evaluate NLMs as "psycholinguistic subjects" – that is, assessing the extent to which their probability distributions conform to human judgments on linguistic data. This psycholinguistic assessment is typically done by evaluating the model on minimal pairs of sentences, which differ only at a target word or phrase that determines the acceptability of the sentence. If an NLM has learned the linguistic phenomenon in question, then it should assign higher probability to sentences that humans judge to be more acceptable. This approach has shown NLMs to be capable of learning some grammatical phenomena (e.g. subject-verb agreement and filler-gap dependencies) while failing on others (Linzen et al., 2016; Lau et al., 2017; Futrell et al., 2018; Gulordava et al., 2018; Marvin and Linzen, 2018; Tran et al., 2018; Wilcox et al., 2018).

In evaluating these mixed learning outcomes, we raise a broader question that remains largely unaddressed in the field: *What is the standard to which we should be holding artificial language models?* An engineering goal within the machine learning community is to build NLMs that approximate human behavior. In this case, an ideal NLM should achieve high performance even on low-frequency constructions, and the learning signal should be detectable even with coarse experimental paradigms. However, if a scientific goal is to highlight the grammatical phenomena that can be learned from sequential data, then experiments should be designed with the aim to give NLMs a fair shot at displaying successful learning.

We demonstrate the value of robust psycholinguistic methods in serving the latter goal by re-evaluating the performance of neural language models on English reflexive anaphor licensing (RAL). For example, in *John disappointed himself*, the reflexive *himself* can refer to *John*, but in *John knew that Paul disappointed himself*, the reflexive can only refer to *Paul* but not *John*. A priori, we expect RAL to be difficult to learn for sev-

[1] Code and data are available at https://github.com/jennhu/reflexive-anaphor-licensing.

eral reasons. From a theoretical perspective, multiple syntactic constraints are simultaneously operative in RAL, which may increase the complexity of the representation that needs to be learned (see Section 2.1). In addition, the appearance of a reflexive is never obligatory based on the preceding context – that is, while a reflexive requires an antecedent NP licensor, an antecedent NP never requires a reflexive downstream (see Section 2.2).

Previous studies have shown NLMs to fail at RAL in various syntactic configurations (Futrell et al., 2018; Marvin and Linzen, 2018). We take a closer look at these previously reported failures, conducting new experiments that control for confounding variables and creating new materials that are compatible with small-vocabulary NLMs. Our experiments detect stronger evidence of learning than reported in previous work, demonstrating the value of robust psycholinguistic methods in studying the potential of NLMs to learn complex syntactic phenomena.

2 Background

2.1 Reflexive anaphor licensing (RAL)

English reflexive anaphors are licensed only when two different structural constraints are both satisfied, which we refer to as LOCALITY and C-COMMAND. These two constraints are independently motivated on theoretical grounds and underlie many syntactic configurations (e.g. Reinhart, 1983; Rizzi, 2013).

LOCALITY stipulates that the matching antecedent must be in the same clause as the reflexive. C-COMMAND requires the matching antecedent to be in a c-commanding relation with the reflexive (Reinhart, 1981; Chomsky, 1993). For present purposes, it is sufficient to define c-command as the following: if a node has any sibling nodes in a syntax tree, then it c-commands its siblings and all of their descendants; if a node has no siblings, then it c-commands everything its parent c-commands.

To illustrate these two constraints, Figure 1 shows the syntax tree for the sentence *The fathers said the women near the boys saw themselves*. This sentence contains three noun phrases (NPs) that could potentially act as an antecedent for *themselves*, but only one of them satisfies both structural requirements of RAL: (1) the higher subject NP_1 *the fathers* c-commands *themselves* but is not within the local clause, violating LOCALITY; (2) the lower subject NP_2 *the women* c-commands *themselves* locally, licensing the reflexive; (3) the linearly closest NP_3 *the boys* is within the local clause, but violates C-COMMAND since it is inside a prepositional phrase inside NP_2. Thus, NP_2 *the women* is the only possible licensor for the reflexive *themselves*.

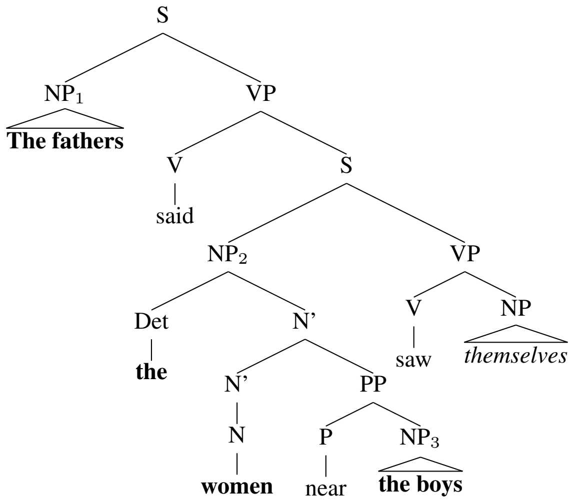


Figure 1: Syntax tree for example sentence. While each NP agrees in number with the reflexive *themselves*, only NP_2 occurs in a position that can license it.

We frame our experiments in terms of the two syntactic constraints involved in RAL, i.e. LOCALITY and C-COMMAND. This is typically done when testing the linguistic knowledge of humans, in order to probe the nature of linguistic generalizations that are being drawn across different types of constructions. In following this convention, we do not intend to claim the NLMs are learning these abstract structural properties per se.

2.2 Distribution of reflexive anaphors

The presence of a reflexive anaphor is never obligatory, in the sense that nothing in the preceding context deterministically predicts an upcoming reflexive. This contrasts with other syntactic dependencies, where the two elements of the dependency mutually require each other. In subject-verb agreement, for example, a subject NP sets the expectation for a downstream verb that agrees in number, and the verb requires a matching subject. This is also the case for less frequent constructions such as filler-gap dependencies, where the appearance of a filler *wh*-word sets the expectation for a gap, and the presence of a gap requires a preceding filler. This property does not hold for reflexive anaphors, as an NP never requires the appearance of a reflexive downstream. Thus, given an upstream reflexive licensor, there is high variance in the downstream contexts.

Furthermore, although we are interested in reflexive anaphors that occur in an argument position, these pronouns can also occur as an intensifier adjoining right next to an NP, as in *The president himself signed my book*. Since the intensifier usage does not obey the same structural constraints, it has a different distribution from the anaphor usage. Both of the factors discussed above pose a challenge for NLMs to learn a robust representation for RAL.

2.3 Paradigms in previous work

Previous work evaluating the ability of neural language models to learn RAL primarily builds upon the paradigms introduced in Marvin and Linzen (2018) and Futrell et al. (2018). Both studies conclude that NLMs fail to learn the appropriate licensing conditions for reflexives.

In particular, Marvin and Linzen (2018) test whether NLMs learn RAL in relative clauses and sentential complements. Consider the following sample items (1) and (2) from their study:

(1) The bankers who the pilot embarrassed hurt *himself / themselves.

(2) The bankers thought the pilot embarrassed himself / *themselves.

In (1), the reflexive *himself* cannot be licensed by *the pilot* because *the pilot* is inside a relative clause, thus violating both LOCALITY and C-COMMAND. In (2), the reflexive *themselves* is embedded in a sentential complement, so the long-distance subject *the bankers* cannot license the reflexive for violating LOCALITY.

As is typical in psycholinguistic evaluation of NLMs, previous RAL studies calculate accuracy as the proportion of trials where the model assigns higher probability to the correct reflexive given the prefix, compared to another reflexive that would make the sentence ungrammatical. Since Marvin and Linzen (2018) and Futrell et al. (2018) test number and gender agreement, respectively, Marvin and Linzen compare the probability of *himself/herself* vs. *themselves*, while Futrell et al. compare the probability of *himself* vs. *herself*.

While the failures reported by these studies have been taken as evidence of the limits of NLM learning, they might be attributed to confounding factors in the design of the experiments. As discussed above, previous studies measure accuracy by comparing the probability assigned to different target reflexives given the same context. However, in many standard training corpora, the reflexive pronouns *themselves*, *himself*, and *herself* differ dramatically in frequency, leading to an asymmetry in unigram probabilities (Table 2). This presents a confound, as all models are likely to implicitly factor unigram probabilities when estimating conditional probabilities in context.[2] Thus, even if a model has learned correct generalizations about the relevant features of the context, these generalizations could be obscured by large differences in unigram frequency.

In addition, both Marvin and Linzen (2018) and Futrell et al. (2018) use profession nouns that are almost all stereotypically male (e.g. *banker*, *senator*). However, many of these nouns occur with low frequency in standard training datasets, so existing materials cannot be used to test RAL learning in models with relatively small vocabularies.

To re-evaluate NLM learning potential of RAL, we conduct new experiments that mitigate the issues raised by unigram probability asymmetries and stereotypically gendered nouns. We describe our methods in Section 3.

3 Experimental design

Psycholinguistic evaluation of language models typically measures accuracy as the proportion of trials in which the model correctly assigns higher probability to the grammatical sentence in a minimal pair. This probability differential is affected not only by the expectations set by the context, but also by the unigram probabilities of the target words (in the case of RAL, *themselves*, *himself*, and *herself*). To avoid this issue, we keep the target reflexive fixed and vary the preceding lexical items in each condition.

3.1 Conditions

Each sentence in our test suites has two NPs, a verb, and a target reflexive, as well as material that modulates the syntactic state (e.g. the onset of a relative clause). One NP is in a position that can license a reflexive, and the other NP is not. Our experiments have the following three conditions:

[2] A unigram frequency is one of the easiest things for a neural model to learn, e.g. as the bias term in the output layer.

- **Baseline:** Both NPs match the number feature of the target reflexive. The sentence is grammatical.

- **Distractor:** The NP in the licensing position matches the number of the target, but the other NP mismatches. The sentence is still grammatical, but contains distracting material.

- **Ungrammatical:** The NP in the licensing position mismatches the number of the target. The sentence is ungrammatical.

We choose to test number instead of gender feature agreement (cf. Futrell et al., 2018) because we believe models are more likely to learn a representation of number than gender, as number is more frequently marked than gender in English. There is also evidence of NLMs learning other number-based dependencies such as subject-verb agreement (Linzen et al., 2016).

3.2 Evaluation metric

Our accuracy calculation involves a three-way comparison. For a given item, the model makes a correct prediction if the probability of the target reflexive in the Ungrammatical condition is lower than the probability of the target in *both* the Distractor and Baseline conditions. Accuracy is the proportion of items in the experiment for which the model makes the correct prediction. If the probability of the target is the same across conditions, then the prediction is considered correct with probability 1/3. Under this measure, chance performance is 33.33%, in contrast to the 50% from existing paradigms that compare grammatical vs. ungrammatical constructions.

3.3 Lexical items

Nouns Previous studies on RAL use nouns denoting professions often associated with stereotypical gender, such as *lumberjack* and *hairdresser* (Futrell et al., 2018; Marvin and Linzen, 2018).[3] However, these nouns are not inherently gendered, and manipulating the gender of the reflexive does not change the grammaticality of the sentence. Instead, we use high-frequency nouns with lexicalized gender, such as *man* and *woman*. This allows us to extend our paradigm to models with smaller vocabularies (see Section 4), for which many profession nouns are out-of-vocabulary (e.g. *hairdresser*). This also ensures that our experiments can be replicated with future corpora, as the stereotypical gender of occupations represented in word embeddings can vary across time and cultures (Garg et al., 2018). We selected a total of 10 nouns (5 female and 5 male), with the female and male nouns balanced for frequency of occurrence in the Wikipedia corpus (see Table 2).

Verbs We first manually constructed a list of commonly used reflexive verbs. Using this list, we calculated the relative frequency of their occurrences within a reflexive construction in the Wikipedia corpus, and selected the most frequent ones. We also selected the most frequent verbs by their raw counts in the corpus. A total of 15 verbs were selected using this method.

Counterbalancing To ensure that vocabulary differences in preceding context do not confound the observed effects on the target reflexive, we counterbalance the position of nouns such that each noun occurs in a licensing and a non-licensing position equally often. Consequently, each stimulus item has several variants, where the nouns are equally distributed across positions. Each noun also appears with each of the verbs equally often across items.

3.4 Logic of experiments

In Experiment 1, we first perform a loose replication of Marvin and Linzen (2018) by adapting their materials into our experimental paradigm. The experiment includes relative clause and sentential complement constructions, which we test in Experiments 1a and 1b, respectively. To construct the materials, we crossed 10 nouns with 7 matrix verbs from the original Marvin and Linzen study, resulting in a total of 70 items per pronoun.

As discussed in Section 2.3, one issue with previous studies is the choice to use lexical items with stereotypical gender. In subsequent experiments, we create new test suites with materials using lexicalized gender. In Experiments 2a and 2b, we use our new materials to test relative clause and sentential complement constructions, respectively, for comparison with Experiments 1a and 1b.

Since the relative clause construction tests both LOCALITY and C-COMMAND and the sentential complement construction only tests LOCALITY, we test prepositional phrases in Experiment 3 to isolate the effect of C-COMMAND. We cross 4

[3] RNNs have been shown to learn NP stereotypical gender (Rudinger et al., 2018).

	Condition	Example sentence
LOCALITY & C-COMMAND		
Relative clause (M&L)	Grammatical	The bankers who the pilot embarrassed hurt themselves
	Ungrammatical	*The bankers who the pilot embarrassed hurt herself
Relative clause (Exp. 1a)	Baseline	The {banker, pilot} that the {pilot, banker} embarrassed hurt herself
	Distractor	The {banker, pilot} that the {pilots, bankers} embarrassed hurt herself
	Ungrammatical	*The {bankers, pilots} that the {pilot, banker} embarrassed hurt herself
Relative clause (Exp. 2a)	Baseline	The {mother, girl} that the {girl, mother} liked saw herself
	Distractor	The {mother, girl} that the {girls, mothers} liked saw herself
	Ungrammatical	*The {mothers, girls} that the {girl, mother} liked saw herself
LOCALITY ONLY		
Sentential complement (M&L)	Grammatical	The bankers thought the pilot hurt herself
	Ungrammatical	*The bankers thought the pilot hurt themselves
Sentential complement (Exp. 1b)	Baseline	The {banker, pilot} said that the {pilot, banker} hurt herself
	Distractor	The {bankers, pilots} said that the {pilot, banker} hurt herself
	Ungrammatical	*The {banker, pilot} said that the {pilots, bankers} hurt herself
Sentential complement (Exp. 2b)	Baseline	The {mother, girl} said that the {girl, mother} saw herself
	Distractor	The {mothers, girls} said that the {girl, mother} saw herself
	Ungrammatical	*The {mother, girl} said that the {girls, mothers} saw herself
C-COMMAND ONLY		
Prepositional phrase (Exp. 3)	Baseline	The {mother, girl} near the {girl, mother} saw herself
	Distractor	The {mother, girl} near the {girls, mothers} saw herself
	Ungrammatical	*The {mothers, girls} near the {girl, mother} saw herself

Table 1: Sample stimuli for *herself* in our experiments and the original Marvin and Linzen ("M&L") study.

nouns with 15 verbs, resulting in 60 items for each pronoun in each of Experiments 2 and 3.[4] Table 1 shows sample items for Experiments 1-3 along with corresponding items from the original Marvin and Linzen (2018) study.

4 Language models

We evaluate RAL in six neural language models, as well as a baseline n-gram model. Together, the models cover a range of vocabulary sizes, architectures, and inductive biases (Table 2). Our goal here is not to draw general conclusions about certain architectures or training regimes, but to present results across a diverse set of models, including those that were previously untestable due to experimental design.

GRNN and JRNN Recurrent neural networks (RNNs; Elman, 1990; Mikolov et al., 2010) perform well in language modeling, with long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997; Sundermeyer et al., 2012) being the most popular variant. We test two LSTMs that differ significantly in vocabulary size and have been shown to learn syntactic dependencies to varying degrees of success. The Gulordava et al. (2018) LSTM ("GRNN") was trained on a subset of English Wikipedia with 90M training tokens. The Jozefowicz et al. (2016) LSTM ("JRNN") was trained on the One Billion Word Benchmark (Chelba et al., 2013). JRNN additionally has convolutional neural network character input embeddings.

Transformer-XL and BERT Next, we test two models based on the Transformer architecture (Vaswani et al., 2017). Transformer-XL ("TransXL"; Dai et al., 2019) reuses the hidden states obtained in previous segments, which facilitates modeling of long-term dependencies. BERT (Devlin et al., 2018) is bi-directional, in that it is trained to predict the identity of masked words based on the preceding and following context.[5] Both models were trained on document-level corpora instead of shuffled sentences: WikiText-103

[4] To counterbalance the position of the nouns, there are 6 variants of each item (2 per condition) for *himself* and *herself*, and 12 variants of each item (4 per condition) for *themselves*.

[5] We use the small, uncased version of BERT ($\text{BERT}_{\text{BASE}}$) with no fine-tuning after the initial pre-training tasks.

Model	Architecture	Training data	Training tokens	Vocab size	*themselves*	*himself*	*herself*
BERT	Transformer	BooksCorpus, Wikipedia	3.3B	30K	-	-	-
TransXL	Transformer	WikiText-103	103M	267K	9K	20K	5K
JRNN	LSTM	1B Word Benchmark	1B	800K	103K	124K	34K
GRNN	LSTM	Wikipedia	90M	50K	10K	17K	4K
TinyLSTM	LSTM	PTB §2-21 (terminals)	950K	23K	114	95	12
RNNG	RNNG	PTB §2-21 (trees)	950K	23K	114	95	12
5-gram	n-gram	Wikipedia	90M	50K	10K	17K	4K

Table 2: Language models evaluated in our experiments, along with raw frequency counts of reflexives in the training data. Pre-training data was not publicly released for BERT.

(Merity et al., 2017) for TransXL, and a combination of BooksCorpus (Zhu et al., 2015) and Wikipedia for BERT. Recent work has shown BERT to perform well on reflexive constructions (Goldberg, 2019).

RNNG and TinyLSTM The last two neural models in our test suite have identical vocabularies but differing inductive biases: a recurrent neural network grammar ("RNNG"; Dyer et al., 2016) and a vanilla LSTM ("TinyLSTM"). Both models were trained on the 1-million-word English Penn Treebank §2-21 (Marcus et al., 1993), but TinyLSTM is only trained on the terminal word sequences, while RNNG is trained on the full annotations, which contain complete constituency parses. This minimal difference allows us to observe the effect of structural supervision, which has been shown to be beneficial in acquiring certain grammatical dependencies (Kuncoro et al., 2017; Wilcox et al., 2019). Crucially, the vocabulary of these models is too small to acommodate the lexical items used in previous RAL studies.

***n*-gram** As a baseline, we test a 5-gram model trained on the same Wikipedia data as GRNN. We use Kneser-Ney smoothing to perform backoff.

4.1 Computing word probabilities

In practice, we calculate accuracy (see Section 3.2) by comparing differentials in log probability space at the target pronoun. To obtain the log probability of word w_i assigned by the LSTMs and Transformer models, we compute

$$\log_2 p(w_i|h_{i-1}), \qquad (1)$$

where h_{i-1} is the model's hidden state before observing w_i. This probability is calculated from the model's softmax activation.

To obtain the log probability of w_i in the RNNG, we follow the method used in Hale et al. (2018). We use word-synchronous beam search (Stern et al., 2017) to find the most likely incremental parses, and sum their forward probabilities to approximate $P(w_1, \ldots, w_{i+1})$ and $P(w_1, \ldots, w_{i-1})$. We use 100 for the action beam size and 10 for the word beam size.

In contrast to the other models in our test suite, BERT is bi-directional. To obtain the log probability of w_i, we first feed BERT a sentence with w_i masked out and obtain the word predictions for the masked position. This gives us a probability distribution over words. In practice, since the target reflexive in our items always occurs directly before the final token '`.`', we do not expect the right context to modulate predictions about the target differently across conditions.

5 Results

5.1 Experiment 1: Marvin and Linzen (2018)

The original materials of Marvin and Linzen (2018) use profession nouns that are stereotypically male. Since these nouns are out-of-vocabulary for RNNG and TinyLSTM, we run this experiment only on the large-vocabulary models (BERT, TransXL, JRNN, GRNN, 5-gram).

Exp. 1a: M&L relative clause We first investigate RAL learning in the relative clause construction (see Table 1). Here, the NP inside the relative clause cannot license the reflexive, as such a relationship would violate both LOCALITY and C-COMMAND. Our design differs from Marvin and Linzen (2018) in that we hold the reflexive anaphor constant while varying the context, with the position of the nouns counterbalanced.

Accuracy scores from the original study and

	BERT	TransXL	JRNN	GRNN	TinyLSTM	RNNG	5-gram
LOCALITY & C-COMMAND							
Relative clause (M&L)	0.80†	–	–	0.55*	–	–	0.50*
Relative clause (Exp. 1a)	0.76 $\pm$ 0.057	0.74 $\pm$ 0.059	0.41 $\pm$ 0.067	0.70 $\pm$ 0.062	–	–	0.33
Relative clause (Exp. 2a)	0.70 $\pm$ 0.067	0.70 $\pm$ 0.067	0.68 $\pm$ 0.068	0.45 $\pm$ 0.073	0.16 $\pm$ 0.053	0.24 $\pm$ 0.062	0.33
LOCALITY ONLY							
Sentential complement (M&L)	0.98†	–	–	0.86*	–	–	0.50*
Sentential complement (Exp. 1b)	0.95 $\pm$ 0.029	0.91 $\pm$ 0.038	0.96 $\pm$ 0.026	1.00 $\pm$ 0	–	–	0.33
Sentential complement (Exp. 2b)	0.98 $\pm$ 0.022	0.92 $\pm$ 0.039	0.97 $\pm$ 0.026	0.99 $\pm$ 0.013	0.82 $\pm$ 0.057	0.88 $\pm$ 0.047	0.33
C-COMMAND ONLY							
Prepositional phrase (Exp. 3)	0.99 $\pm$ 0.008	0.71 $\pm$ 0.067	0.69 $\pm$ 0.063	0.75 $\pm$ 0.063	0.43 $\pm$ 0.072	0.62 $\pm$ 0.071	0.33

Table 3: Accuracy scores for each experiment, with 95% confidence intervals shown below where applicable. Accuracy is computed at the item-level for each pronoun, then averaged across all pronouns. Chance accuracy is 33.33%, except for entries marked with † or *, where chance is 50%. The BERT results marked with † come from Goldberg (2019), while the GRNN and 5-gram results marked with * come directly from Marvin and Linzen (2018). These results are also not directly comparable to each other due to the bi-directionality of BERT; see Goldberg (2019) and Wolf (2019) for details.

our Experiment 1 are reported in Table 3 (top two rows). Accuracy is computed at the item-level for each pronoun, then averaged across all pronouns. Under our evaluation method, GRNN shows considerable improvement over what was reported in Marvin and Linzen (2018), while the 5-gram model remains at chance. While our metrics are not strictly comparable, the original study reports near-chance accuracy (55% $\sim$ 50%), while we report accuracy well above chance (70% $\gg$ 33.33%). BERT achieves slightly lower accuracy under our paradigm than was reported in Goldberg (2019) (76% vs. 80%); note, however, that our chance baseline is lower.

Exp. 1b: M&L sentential complement Next, we investigate RAL learning in the sentential complement construction. Here, the long-distance subject cannot license the reflexive embedded in a sentential complement, because such a relationship would violate LOCALITY (while satisfying C-COMMAND). As in Exp. 1a, our approach differs from Marvin and Linzen (2018) in that we hold the reflexive anaphor constant while varying the context, with the position of the nouns counterbalanced.

All large-vocabulary neural models perform near ceiling in our paradigm, despite our metric having a lower baseline. GRNN achieves 100% accuracy, showing a marked improvement over previously reported results (Table 3). Overall, the models exhibit the correct trend for the sentential complement construction (Exp. 1b), but the pattern is less clear for the relative clause construction (Exp. 1a). One possible explanation is that in a relative clause, the licensing NP is linearly farther away from the reflexive than the distracting NP; a global preference for linear proximity may have obscured learning of structural adjacency.

5.2 Experiment 2

The materials used in Marvin and Linzen (2018) (and our Experiment 1) involve items with stereotypically gendered nouns. This raises two potential issues: (1) gender biases may overshadow number mismatch effects, and (2) the materials can only be used to evaluate models with reasonably large vocabularies. As in Experiment 1, the design of Experiment 2 differs from Marvin and Linzen (2018) in that we hold the reflexive anaphor constant while varying the context. In addition, we create new materials using nouns with lexicalized gender rather than stereotypical gender. This allows us to evaluate all seven models in our test suite.

Exp. 2a: Relative clause As in Exp. 1a, we first test RAL learning in the relative clause construc-

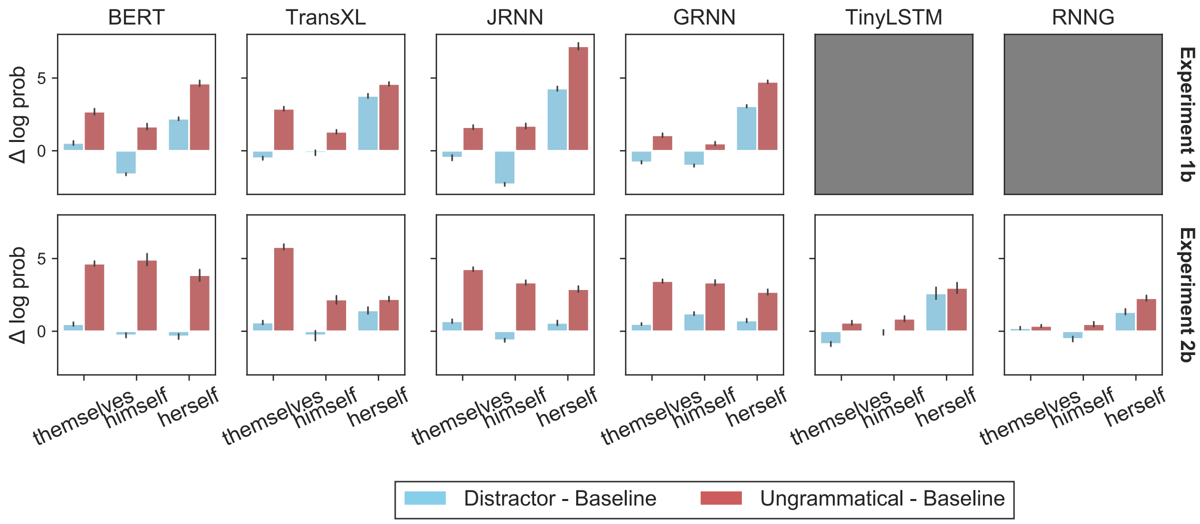


Figure 2: Negative log probability differential at target reflexive in sentential complement construction. Error bars are bootstrapped 95% confidence intervals. **Blue bars:** Distractor-Baseline differential at target reflexive. **Red bars:** Ungrammatical-Baseline differential at target reflexive. If the models learn the correct generalization for RAL, then the red bars should be both positive and higher than the blue bars. **Top (Exp. 1b:)** Distractor-Baseline differential is significantly higher at *herself* than *himself* or *themselves*. The stimuli contain materials that are out-of-vocabulary for TinyLSTM and RNNG. **Bottom (Exp. 2b):** For the large-vocabulary models, the Distractor-Baseline differential is comparable across pronouns. For the small-vocabulary models, the differential is significantly higher at *herself*.

tion using our new set of materials. Accuracy scores are high for most of the large-vocabulary neural models (BERT, TransXL, JRNN) and above chance for GRNN, but at or below chance for the other models (Table 3).

Exp. 2b: Sentential complement In Experiment 3, we test the sentential complement construction using our materials. As shown in Table 1, we place the reflexive inside a complement clause, such that either both c-commanding NPs match the number feature of the reflexive (Baseline), or there is one mismatching NP either in the non-local subject position (Distractor) or the local subject position (Ungrammatical).

All large-vocabulary neural models perform near ceiling (Table 3). The small-vocabulary models RNNG and TinyLSTM achieve lower accuracy, but RNNG outperforms TinyLSTM.

5.3 Experiment 3

Since previous studies have focused on the relative clause and sentential complement constructions, C-COMMAND has not been tested separately from LOCALITY. In Experiment 3, we hold LOCALITY constant while manipulating C-COMMAND by placing a distractor NP inside a non-c-commanding PP modifier in the local subject NP. No clausal boundary is introduced. As in Experiment 2, our approach differs from Marvin and Linzen (2018) in that we hold the reflexive anaphor constant while varying the context, and we use nouns with lexicalized gender.

Accuracy scores are reported in the bottom section of Table 3. Performance is well above chance for all neural models except TinyLSTM. RNNG shows a clear advantage over TinyLSTM (62% vs. 43%).

5.4 Asymmetry between *himself* & *herself*

Thus far, we have reported accuracy scores averaged across the three reflexive pronouns (Table 3). The three pronouns are weighted equally in the reported numbers, as accuracy is computed at the level of each item.

Next, we investigate differences in performance across reflexive anaphors. Figure 2 shows the results of this cross-pronoun comparison for Experiments 1b and 2b, which both use the sentential complement construction (LOCALITY only). Blue bars show the Distractor-Baseline log probability differential at the target reflexive. Red bars show the Ungrammatical-Baseline log probability differential at the target reflexive. If the models learn the correct generalization for RAL, then the red bars should be both positive (i.e. above baseline) and higher than the blue bars.

In Experiment 1b, which uses profession nouns that are primarily associated with men,[6] the Distractor-Baseline differential (blue bars) is significantly higher at *herself* than at *himself* or *themselves*. In contrast, in Experiment 2b, which uses nouns with lexicalized gender, there is only a significant difference between the Distractor-Baseline differentials at *himself* and *herself* for the small-vocabulary models TinyLSTM and RNNG.

We hypothesize that this can be attributed to the choice of vocabulary items. In the Distractor condition of Experiment 1, the distracting noun is plural and has stereotypically male gender (e.g. *senators*). The features of this noun partially match with *himself* (in stereotypical gender but not number), but match in neither feature with *herself*, leading to a higher Distractor-Baseline differential for *herself*. This is not an issue in Experiments 2 and 3, where all nouns match in gender feature with the target reflexive across conditions. However, training data with a low number of occurrences of *herself* can still lead to a high Distractor-Baseline differential, as is the case in Experiment 3 for TinyLSTM and RNNG.

This pattern may also result from a more general asymmetry between gender stereotypes: encountering *herself* after a stereotypically male noun is more surprising than encountering *himself* after a stereotypically female noun. Interestingly, asymmetry also manifests in human production biases, where gendered pronoun production and interpretation are not mutually calibrated (Boyce et al., 2019).

6 Discussion

In this paper, we used new experiments to re-evaluate the performance of neural language models on reflexive anaphor licensing. Our methods address issues in previous studies, such as unigram probability asymmetries between target pronouns and the choice to use nouns with stereotypical gender, which may have led to an underestimation of learning signal. The results suggest that NLMs are learning more about RAL than they have previously been given credit for, and demonstrates the value of robust psycholinguistic methods in highlighting the potential of NLMs to learn complex syntactic phenomena.

[6] 11 out of these 12 nouns are stereotypically male according to United States Census data (Bureau of Labor Statistics, 2017).

The value of our approach extends beyond RAL. If we seek to understand the linguistic generalizations that NLMs can *potentially* acquire, then we must design our experiments to give NLMs a fair shot at displaying successful learning, regardless of the phenomenon under study.

Of course, the generalizations acquired by NLMs may not be well characterized in linguistic terms such as LOCALITY and C-COMMAND, but rather properties of the data that are irrelevant to structural considerations. Further experiments will be required to deepen our understanding of the generalizations underlying the successes and failures of these models on this and other evaluation tasks. More generally, future work in this domain should carefully address hypotheses about language learning, keeping in mind complementary questions that arise from engineering and scientific agendas.

Acknowledgments

We would like to thank Tal Linzen, Peng Qian, and the anonymous reviewers for their insightful comments. J.H. is supported by an NSF Graduate Research Fellowship.

References

Veronica Boyce, Titus von der Malsburg, Till Poppels, and Roger Levy. 2019. Remember 'him', forget 'her': Gender bias in the comprehension of pronomial referents. In *32nd Annual CUNY Conference on Human Sentence Processing*.

Bureau of Labor Statistics. 2017. Labor force statistics from the current population survey.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google.

Noam Chomsky. 1993. *Lectures on government and binding: The Pisa lectures*. 9. Walter de Gruyter.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. *CoRR*, abs/1602.07776.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *CoRR*, abs/1809.01329.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan R. Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 5:1202–1247.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. In *Transactions of the Association for Computational Linguistics*, volume 4, pages 521–535.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of ICLR*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048, Makuhari, Chiba, Japan.

Tanya Reinhart. 1981. Definite NP anaphora and c-command domains. *Linguistic Inquiry*, 12(4):605–635.

Tanya Reinhart. 1983. *Anaphora and semantic interpretation*. Routledge.

Luigi Rizzi. 2013. Locality. *Lingua*, 130:169–186.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700. Association for Computational Linguistics.

Martin Sundermeyer, Ralf Schluter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about fillergap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballestros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf. 2019. Some additional experiments extending the tech report "Assessing BERT's syntactic abilities" by Yoav Goldberg. Technical report, HuggingFace, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Washington, DC. IEEE Computer Society.

Extending the Autosegmental Input Strictly Local Framework: Metrical Dominance and Floating Tones *

Yuhong Zhu
Department of Linguistics
The Ohio State University
Columbus, OH 43210, USA
zhu.1447@osu.edu

Abstract

This paper extends the empirical coverage of the Autosegmental Input Strictly Local (A-ISL) framework (Chandlee and Jardine, 2019) by analyzing three tonal processes: floating tone suffixation in Cantonese, metrical dominance effect in Shanghai Chinese, and a combination of floating tones and metrical dominance in Suzhou Chinese. I show both the adequacy and inadequacy of the current A-ISL framework: it locally resolves some tonal processes that are otherwise non-local (Shanghai), but fails to account for other empirical data due to a lack of tonal membership specification (Suzhou). With the addition of a morphological affiliation tier, I propose an analysis for the Suzhou data. The paper contributes to our typological knowledge of computational locality and autosegmental phonological representations.

1 Introduction

In this paper I aim to build on the *autosegmental input strictly local* (A-ISL) functions introduced in Chandlee and Jardine (2019), and examine some tonal processes not discussed by the previous A-ISL accounts, extending the empirical coverage of A-ISL transductions. I will assess three phonological processes involving tones: floating tone suffixation in Cantonese (Chen, 2000; Yip, 2002), metrical dominance effect in Shanghai Chinese (Duanmu, 1999), and a combination of floating tones and metrical dominance in Suzhou Chinese (Shi and Jiang 2013, my fieldwork).

I show through the examination of these three cases that the A-ISL model is well-equipped to capture most patterns involving floating tones (Cantonese) and metrical dominance (Shanghai). However, the *combination* of floating tone representation and metrical dominance (Suzhou) cannot be A-ISL: The Suzhou tone sandhi map is not definable through *quantifier-free* (QF) *first-order* (FO) logical transductions Chandlee and Lindell in prep, even if we adopt autosegmental instead of linear tonal representation. This particular insight has been discussed in Chandlee and Jardine (2019): although autosegmental phonology is claimed to be a 'solution' to non-local phonological processes as it makes tonal relationship local (Odden, 1994), the *locality* of autosegmental representations does not always hold when evaluated mathematically. I also show that the Suzhou data can be accurately accounted for with a minimal addition of morphological affiliation relations (i.e. association between tones and morphemes).

The paper is structured as follows. In §2 I layout some background on both autosegmental representations and A-ISL transductions. I introduce the three tonal processes and the attempts to analyze them using A-ISL transductions in §3. §4 summarizes the results and proposes an alternative analysis of Suzhou. §5 concludes this paper.

2 Background

2.1 Autosegmental Phonology

Autosegmental Phonology (Goldsmith, 1976) proposes separate autosegmental *tiers* and *association relations* between tiers as part of phonological representation to account for many long-distance/non-local phonological processes. A well-known segmental example is blocking and transparency effects of nasal harmony. Examples of two observed patterns of nasal harmony are given below.

(1) Blocking effect in Johore Malay (Onn, 1980)
/pəŋawasan/ ↦ [pəŋãw̃ãsan] (*[pəŋãw̃ãsãn])

* I would like to thank Jeffrey Heinz, Jonathan Rawski, Jane Chandlee and the anonymous reviewers for their insightful feedback. All errors are my own.

(2) Transparency effect in Tuyuca (Barnes, 1996)
/mipi/ ↦ [m̃ĩp̃ĩ] (*[m̃ĩpi])

In (1) obstruents and liquids block rightward nasalization, whereas in (2) obstruents are *transparent* to the nasal harmony. The difference between these two patterns is often captured by a (under)specified [Nasal] feature of the obstruents on the autosegmental tier: an obstruent that blocks nasalization is necessarily specified with [-Nasal], while one that allows nasality to 'pass through' is underspecified on the Nasal autosegmental tier. Autosegmental representations resolve the non-local nature of such harmony patterns by proposing that relevant features to harmony are still *local* on their respective autosegmental tiers.

For similar reasons, autosegmental representations are useful tools when analyzing tonal processes. Chandlee and Jardine (2019) have evaluated the computational properties of multiple spreading and deletion processes of tones, assuming autosegmental representations. In this paper, I follow their methods and explore three slightly more complex tonal processes than those analyzed in Chandlee and Jardine's paper, drawing data from Chinese languages: The first case (Cantonese) discusses floating tone affixes; the second case (Shanghai) introduces a metrical dominance effect to the A-ISL model; the third case (Suzhou) combines both metrical dominance and floating tone representations.

2.2 Computational preliminaries

All preliminaries come from definitions in Chandlee and Jardine (2019) . For strings, I assume the following in this paper:

(3) a. Σ: A finite alphabet of symbols.
b. Σ^*: Set of all strings over Σ.
c. Strings w, v and their concatenation wv; set of strings L and concatenation between strings and sets of strings wL.

For models, I assume:

(4) a. A model $\langle D|f_1, ..., f_n, R_1, ..., R_m\rangle$ where D is a finite domain of elements, $f_1, ..., f_n$ are a set of functions over the domain, and $R_1, ..., R_m$ are a set of relations over the domain.
b. For our purpose of examining strings, I assume models of the signature $\{p, s, P_{\sigma\in\Sigma}\}$.
c. p, s: *predecessor* and *successor* functions. $p(i) = i - 1$, $s(i) = i + 1$, with the exceptions that the first element is its own predecessor ($p(1) = 1$) and the last element is its own successor ($s(n) = n$ for a string of length n).
d. $P_{\sigma\in\Sigma}$: a unary relation for every $\sigma \in \Sigma$ that gives the label of each position of the string.
e. A user-defined function first(x): $\texttt{first}(x) \overset{\text{def}}{=} p(x) = x$.
f. A user-defined function second(x): $\texttt{second}(x) \overset{\text{def}}{=} (\neg p(x) = x) \wedge (p(p(x)) = p(x)$.
g. A user-defined function last(x): $\texttt{last}(x) \overset{\text{def}}{=} s(x) = x$.

I follow Chandlee and Jardine (2019) in using QF (Quantifier-Free) logic: For all QF formulae $\psi(x_1, ..., x_n)$, the variables $x_1, ..., x_n$ are unbounded by quantifiers. For logical transductions, I assume:

(5) a. An input model signature I, an output model signature O.
b. $\psi(x)$: a unary predicate in the input I.
c. For each function $f \in O$, $f(x) \overset{\text{def}}{=} \psi_f(x, y)$ for some $\psi_f(x, y)$ in I.
d. For each unary relation $P \in O$, $P \overset{\text{def}}{=} \psi_P(x)$ for some $\psi_P(x)$ in I.
e. For each binary relation $R \in O$, $R \overset{\text{def}}{=} \psi_R(x, y)$ for some $\psi_R(x, y)$ in I.
f. $M \models \psi(x_1, ..., x_n)$: the model M *satisfies* $\psi(x_1, ..., x_n)$. For each set of $x_1, ..., x_n$ in M the formula $\psi(x_1, ..., x_n)$ is evaluated to be true. This defines *n-ary* mappings between input and output.

A logical transduction τ maps models of input structure M_I to those of output structure M_O, where:

(6) a. For each $x \in D$ there is a copy x' in the output iff $M_I \models \psi_D(x)$.
b. For some pair $x, y \in D$ and for each function $f(x') \in O$, there is a copy pair x', y' in the output such that $f(x') = y'$ iff $M_I \models \psi_f(x, y)$.
c. For some $x \in D$ and for each unary rela-

tion $P \in O$, there is a copy $x' \in P$ in the output iff $M_I \models \psi_P(x)$

d. For some pair $x, y \in D$ and for each binary relation $R \in O$, there is a copy pair $(x', y') \in R$ in the output iff $M_I \models \psi_R(x, y)$

2.3 Autosegmental models

Segmental information of strings is irrelevant for the purpose of the current paper and thus will be omitted. I will assume a Tone-Bearing-Unit (TBU) tier and a Tonal tier (containing High, Mid and Low tones) for the rest of this paper. The TBU tier and the tonal tier are treated as separate strings, connected by a binary association relation. A sufficient model signature for autosegmental representations is presented in (7):

(7) $\langle D | p, s, A, P_H, P_M, P_L, P_\sigma \rangle$

Where D is the domain, p, s are predecessor and successor functions, A is a binary association relation between tones and TBUs, P_H, P_M, P_L are unary relations for High, Mid and Low tones, and P_σ is a unary relation for TBUs (syllables). With respect to contour tones, I follow general autosegmental representations (Yip 2002 for discussion) and treat them as sequences of level tones (i.e. a high falling tone is represented as a HL sequence; see §3.2). Moreover, concatenation of autosegmental representations will simply be concatenations of strings on each autosegmental tier, preserving all association relations in A.

I will include examples for each of the cases examined in the following section. For now, I will demonstrate the tonal map process with an artificial bounded deletion example.

(8) Bounded tone deletion
/mòmó/ ↦ [mò.mo]

The process in (8) can be captured by the A-ISL model in Figure 1:

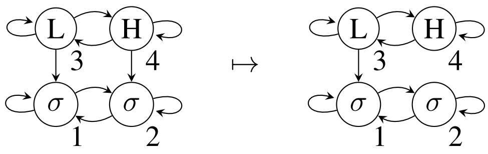


Figure 1: An example of autosegmental tonal mapping

As shown, the process of final tone deletion can be captured as the deletion of a tone-TBU association relation: the association between H (position 4) and the last syllable (position 2) — A(4,2) — is deleted on the output[1]. This hypothetical tone deletion process is definable through the following QF logical formulae, and is this A-ISL:

(9) a. $\sigma'(x) \overset{\text{def}}{=} \sigma(x)$
b. $T'(y) \overset{\text{def}}{=} T(y)$
c. $A'(y, x) \overset{\text{def}}{=} A(y, x) \wedge \neg \texttt{last}(y)$

(9a) states that the unary relation $\sigma'(x)$ is true if $\sigma(x)$ is true. This in turn maps all TBUs in the input (1 and 2) faithfully to the output. (9b) similarly maps all tones from the input to the output (I use T here as a short hand for all H/M/L tones). (9c) defines the crucial tonal process: the binary association $A'(y, x)$ is true if both of the following are true: (i). $A(y, x)$ is true; (ii). $\texttt{last}(y)$ is false. Put plainly, input tonal association lines are preserved in the output only if the tone is *not* the last element on the Tonal tier (in this case, 4).

In the next section, I show that the current A-ISL model (i) achieves the same empirical coverage of the ISL model in representing floating tone affixation; (ii) is able to resolve a crucial case that is non-ISL if viewed linearly (Shanghai); (iii) is unable to account for the combination case (Suzhou) due to model-external reasons.

3 Floating tones and metrical dominance

3.1 Floating tone suffixation in Cantonese

Both Chen (2000) and Yip (2002) present a case of the 'familiar vocative' affix in Cantonese as a demonstration of floating tone suffixation. The relevant data is presented in (10):

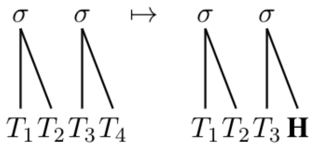
(10) a. [a](M) 'Old', a vocative prefix
b. [tsæng](HM) 'Zhang', a last name
c. [ts^{h}an] (ML) 'Chen', a last name
d. [a.tsæng] (M.HH) 'Old Zhang'
e. [a.ts^{h}an] (M.MH) 'Old Chen'

The process is rather straightforward: a floating H morpheme is attached to the right edge of the familiar vocative term, overwriting the rightmost tone of the rightmost syllable (Chen, 2000; Yip,

[1]Tone deletion can also be captured by the deletion of tonal elements themselves. For our current purposes, I will assume that deletion of association lines achieves the same effect, as unassociated tones are not pronounceable on the surface (Yip, 2002).

2002). I will treat the process as final tone *substitution* instead of concatenation then deletion[2]:

(11) Cantonese H tone suffixation



Interestingly, this process is both ISL and A-ISL.

(12) *Cantonese floating H suffixation is ISL.* Assume a linear transformation $\mathrm{T}_1...\mathrm{T}_k \mapsto \mathrm{T}_1...\mathrm{T}_{k-1}\mathrm{H}$ for any k, two input strings $\mathrm{T}_1...\mathrm{T}_k$ and $\mathrm{T}_0\mathrm{T}_1...\mathrm{T}_k$ have the same k-suffix ($\mathrm{T}_1...\mathrm{T}_k$). Moreover, an input extension $\mathrm{T}_k + 1...\mathrm{T}_n$ to the two strings will result in the same output contribution: $\mathrm{T}_k...\mathrm{T}_{n-1}\mathrm{H}$. The two strings have the same *tails* (see the formal definition of tails in Chandlee 2014).

This process is A-ISL as it is QF-definable by the following transduction:

(13) a. $\sigma'(x) \stackrel{\text{def}}{=} \sigma(x)$
b. $H'(y) \stackrel{\text{def}}{=} H(y) \vee \texttt{last}(y)$
c. $M'(y) \stackrel{\text{def}}{=} M(y) \wedge \neg\texttt{last}(y)$
d. $L'(y) \stackrel{\text{def}}{=} L(y) \wedge \neg\texttt{last}(y)$
e. $A'(x, y) \stackrel{\text{def}}{=} A(x, y)$

In the above formulae, x represents TBU elements and y tonal elements. Shown in (13), two input-output mappings are identical copies: (13a) faithfully maps input TBUs to the output, where (13e) preserves all association relations. The H tone substitution process is defined through the tonal mappings (13b)-(13d): a tone is a H tone in the output if it is H in the input *or* it is the last tone; it is a M/L in the output if it is M/L in the input *and* it is *not* the last tone. An A-ISL model demonstration of [a.tsʰan] (M.MH) 'Old Chen' is given below (predecessor and successor relations are omitted for readability).

Figure 2 illustrates the tonal mapping /M.ML/ ↦ [M.MH]. The unary relations for TBUs and the binary relations for tone-TBU associations are

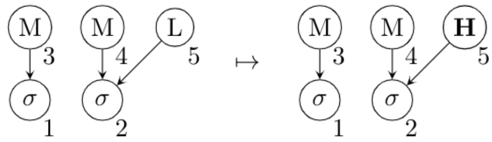


Figure 2: Cantonese H tone suffixation: A-ISL map

kept constant from input to output. The two M tones (in 3 and 4) are also mapped faithfully. Crucially, the tonal element in position 5 satisfies (13b) and does not satisfy (13d). As a result, a L tone in the input is substituted with a H tone in the output for 5.

One fact further complicates the Cantonese data: if the rightmost syllable has a level tone, the floating H affixation process will create a contour tone instead of overwriting the rightmost H level: /M.L/ ↦ [M.LH]. It requires a bit more effort for the tonal map to differentiate level or contour tones. However, changing the representation of L level to LL[3] correctly accounts for the transformation without altering the transduction itself.

3.2 Left dominance in Shanghai tone sandhi

Shanghai is a variety of Northern Wu Chinese, well known for its distinctive tone sandhi patterns. The relevant tone sandhi data for our concern is given below (data from Duanmu 1999; tones in parentheses are surface tones):

(14) a. [ŋ̍] (LM) 'fish'
b. [ɕʲo] (MH) 'small'
c. [wã] (LM) 'yellow'
d. [ɕi] (HM) 'fresh'
e. [ɕʲo.ŋ̍] (M.H) 'small fish'
f. [wã.ŋ̍] (L.M) 'yellow fish'
g. [ɕi.ŋ̍] (H.M) 'fresh fish'
h. [ɕʲo.wã.ŋ̍] (M.H.L) 'small yellow fish'
i. [ɕi.wã.ŋ̍] (H.M.L) 'fresh yellow fish'
j. [ɕʲo.ɕi.wã.ŋ̍] (M.H.L.L) 'small fresh yellow fish'

A few generalizations can be made: first, only monosyllabic words carry contour tones;[4] second, tonal material of the initial syllable seems to be re-

[2]This process can be interpreted as tonal substitution precisely due to the fact that the floating H affix is without segmental information. A process requiring tone-segment affiliations but not tone-TBU associations (i.e. floating tones *with* segments) is challenging to the current A-ISL model. See the case of Suzhou in 3.3.

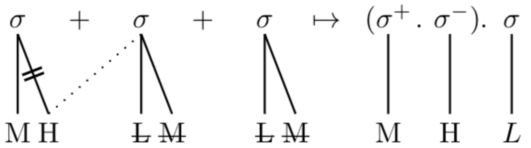
[3]This could be motivated by proposing that Cantonese is a *mora-TBU* language, and unreduced syllables in Chinese languages are usually bimoraic.

[4]Duanmu (1999) accounts for this by proposing that all syllables in Shanghai are underlyingly monomoraic, and they only get lengthened to be bimoraic when in isolation. Consequently, only syllables with two moras can carry two level tones, contributing to a contour.

tained (and 'redistributed') in polysyllabic words; lastly, the third and fourth syllables surface as L.

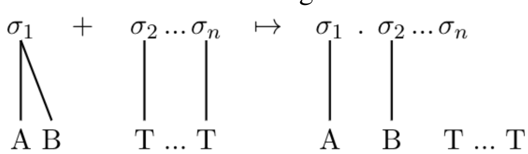
Duanmu's (1999) analysis of Shanghai tone sandhi patterns proposes a metrical 'left dominance' effect. Simply put, footing in Shanghai is left-to-right, non-recursive and trochaic, giving phonological prominence[5] to *only* the initial syllable in a prosodic word. According to Duanmu (1999), the initial syllable is the foot head and always retains its tonal material in tone sandhi positions. The second syllable, being the foot dependent, loses all of its tones. Additionally, tonal material from the initial syllable is shared between the first two (footed) syllables by tonal reassociations. Any unfooted syllables (third, fourth...) loses all tones and surface as toneless L (I will use italic L to represent any phonetic L tones from phonologically toneless syllables). An autosegmental demonstration of the process is given in (15):

(15) An autosegmental derivation of [ɕʲo.wã.ŋ̩] 'small yellow fish'. σ^+ stands for a footed head syllable and σ^- a footed dependent. The third syllable is phonologically toneless and surfaces as a phonetic L.



Abstracting away from the language data, the left dominant tone sandhi process can be represented as deletion and addition of association lines while keeping all tones intact — I assume that tones without association to TBUs (i.e. floating tones) are not pronounced in the output[6]. Phonologically toneless syllables in the output are then subject to surface phonetic implementation (phonetic L in Shanghai).

(16) Left dominance in Shanghai



[5]Being in the metrical head position does not necessarily entail phonetic stress (increased intensity and duration, higher pitch). The metrical prominence here could be purely phonological in that it does not have any phonetic correlates.

[6]A transduction with vowel deletion and reassociation would be indistinguishable from the current map in the output.

In (16), A and B stand for underlying tones of the first syllable, T stands for any tones (contour or level). This tone sandhi process is not ISL: A transformation assuming strings of tones $\mathrm{ABT}^n \rightarrow \mathrm{AB}\emptyset^{n-1}$ does not reflect the fact that tonal material of the initial syllable is redistributed between the first two (footed) syllables. This is directly caused by my representation of contour tones as subsequent level tones associated to one TBU — Contour tones can be 'broken apart' and shared between two syllables in Shanghai. Therefore, it is not possible to represent them as standalone units (e.g. R for Rise; see Chandlee (2018) on tone sandhi in Tianjin). Consequently, it is necessary to adopt autosegmental representations since contour tones entail many-to-one tone mapping.[7]

One reviewer has suggested that including syllable boundary symbols [.] in the alphabet could potentially resolve the non-locality of the Shanghai map: /AB.CD.EF/ $\mapsto$ [A.B.$\emptyset$]. This is indeed correct for the synchronic data of Shanghai: every lexical tone in contemporary Shanghai is a contour by historical coincidence. On the other hand, several neighboring dialects of Shanghai have complex contours (i.e. monosyllables with three tones), and it is logically possible for a syllable to have more than three tonal elements (albeit being typologically unattested). Since the existence of complex contours cannot be ruled out for theory-internal reasons, an ISL map should be able to handle tonal input of *indefinitely many tones within a syllable*. This is shown in (17)

(17) Hypothetical left-dominant mapping with complex contour input
$/\mathrm{T}_1\mathrm{T}_2...\mathrm{T}_n.\mathrm{T}_{n+1}/ \mapsto [\mathrm{T}_1.\mathrm{T}_2]$

This map is Non-ISL, because the transduction needs to 'remember' the second tone $/\mathrm{T}_2/$ for an indefinite length until it encounters the first syllable boundary. 'Remembering' in ISL is achieved through a finite k-factor window (Chandlee, 2014). The current map with an indefinite memory length cannot be ISL.

However, the tone sandhi process in Shanghai is A-ISL since we can easily define it with

[7]Strictly speaking, Shanghai tone sandhi is non-ISL not because of any property of ISL functions, but because of the nature of *linear* phonological transformations: any current phonological framework without an autosegmental representation of contour tones will have a hard time accounting for these data.

a quantifier-free transduction using autosegmental representations.

(18) a. $\sigma'(x) \overset{\text{def}}{=} \sigma(x)$
b. $H'(y) \overset{\text{def}}{=} H(y)$
c. $M'(y) \overset{\text{def}}{=} M(y)$
d. $L'(y) \overset{\text{def}}{=} L(y)$
e. $A'(y,x) \overset{\text{def}}{=} (A(y,x) \wedge \texttt{first}(x) \wedge \texttt{first}(y)) \vee (A(y,p(x)) \wedge \texttt{second}(x) \wedge \texttt{second}(y))$[8]

The formulae (18a)-(18d) preserve *all* input TBUs and tones in the output. The formula in (18e) states the association relations in the output structure: a TBU is associated with a tone in the output if: (i). there is an association between it and the first tone in the input, and it is the first TBU; (ii). there is an association between *its predecessor TBU* and the second tone in the input, and it is the second TBU. I demonstrate the A-ISL map using the example [ɕi.ŋ̍] (H.M) 'fresh fish'.

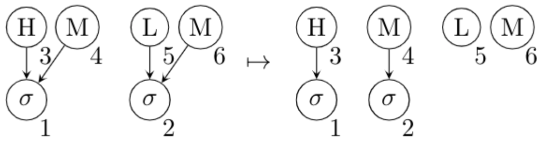


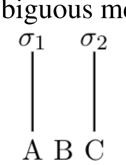
Figure 3: Shanghai left-dominant sandhi: A-ISL map

In Figure 3, all TBUs and tones (1 through 6) are preserved in the output. The autosegmental relation $A(3,1)$ satisfies the left disjunct of (18e) (first tone to first TBU), and is mapped faithfully to the output. The right disjunct of (18e) is satisfied when x is 2 and y is 4 (predecessor TBU is associated with the second tone), therefore a new autosegmental association $A'(4,2)$ is established in the output. As discussed above, left-over tones without association lines (5 and 6) are not pronounced in the output.

Interestingly, this transduction handles situations where the initial syllable only has one tone correctly as well: /H/ + /ML/ $\mapsto$ [H.*L*]. As there is not a pair of value that satisfies the right disjunct of (18e), the second syllable will not be associated with any tones in the output and becomes toneless. The same is true with situations where the initial syllable has *indefinitely many tones*: the second tone will be displaced, whereas all left-over tones remain floating.

A related observation is that tones in Shanghai show their 'membership' status through the association relations: the first two tones 'belong' to the initial syllable, because they are *associated* with the initial syllable. If morphemes in the language contain floating tones, our current model is not able to determine its affiliation status. This is demonstrated in (19):

(19) An autosegmental representation with ambiguous membership status



The current autosegmental model has no way of expressing morphological affiliation of floating tones: we know σ_1 precedes σ_2, and tone B is in between tones A and C. However, there is no way to determine if the floating tone B comes from σ_1 or σ_2 underlyingly. This poses a problem when we encounter languages utilizing *both* metrical dominance and floating tones (see §3.3).

3.3 Floating tones and left dominance in Suzhou tone sandhi

The tone sandhi data of Suzhou Chinese comes from (Shi and Jiang, 2013) and my fieldwork (Zhu, in prep). Here, I present two pairs of alternation that motivate *both* floating tone representation and left dominance:

(20) a. [sʲæ] (HL) 'small'
b. [mã] (LH) 'blind'
c. [nɪn] (LH) 'person'
d. [sʲæ.nɪn] (HL.*L*) 'child'
e. [mã.nɪn] (L.H) 'blind person'

Left dominance is still present in Suzhou: tones from initial syllables are always preserved in polysyllabic words, while tones from non-initial syllables are all deleted. Crucially, tonal redistribution does not always take place in Suzhou: in (20d), an underlying /HL/ falling tone stays in the initial syllable, where as in (20e) the underlying /LH/ evenly

[8]One reviewer has expressed concerns with the user-defined function second(x): why are there only first and second, but not third and more? This echoes with the insightful observation made by (Kenstowicz, 1994): 'linguistic rules do not count beyond two'. Here, the binary tone pattern in Shanghai is accounted for using a non-iterative binary foot (Duanmu, 1999). Including larger prosodic constituents (e.g. feet, prosodic words) is perfectly in line with principles of Autosegmental Phonology and the current A-ISL model. As such, a metrical foot level is not yet incorporated purely due to space constraints rather than model-internal limitations.

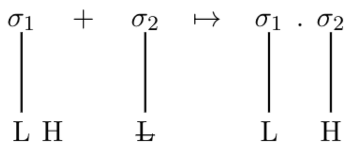
distributes itself over two syllables. Such pattern is inherent to the two lexical tones and is systematic across different morphemes.

To account for the different movability nature of the two lexical tones, I propose a tonal representation contrast between (20a) and (20b): Both H and L are associated in (20a), while (20b) has both tones floating underlyingly. The autosegmental representations are given in (21):

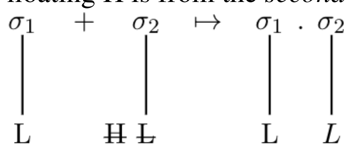
(21) a. ‘small’ b. ‘blind’

The tone sandhi process in Suzhou differs from that of Shanghai in that associated tones cannot be redistributed to other TBUs (due to a general restriction on deleting association lines). Only floating tones from the initial syllable can be freely associated to other footed syllables in sandhi position. This process is not ISL for the same reason given in 3.2: one linear string of tones cannot express many-to-one tonal association relations. Moreover, this process is also Non-A-ISL. Since floating tones have no way to express their morphological affiliation status under the current autosegmental representations (recall (19)), the model cannot determine if a floating tone belongs to the initial syllable or not — left dominance cannot function on floating tones under the current framework. Consider a more concrete pair of examples in (22) and (23):

(22) A LH.L disyllabic sequence where the floating H is from the *initial* syllable.



(23) A L.HL disyllabic sequence where the floating H is from the *second* syllable.



In (22), the sequence contains an initial LH syllable with a floating H and a second L syllable. The floating H tone is redistributed to the second syllable in the output. In contrast, (23) contains an initial L syllable and a second HL syllable with a floating H. Both tones in the second syllable are deleted in tone sandhi, and the surface form would be [L.L] instead of [L.H]. Our current framework cannot differentiate (22) and (23), since there is no way to express membership of floating tones.

4 Discussion

4.1 Evaluation of analyses

In §3 I have illustrated three tonal processes in Chinese and their ISL/A-ISL status. The result is summarized in Table 1 below.

I have shown three out of four different logical possibilities of ISL and A-ISL transductions: (i). both ISL and A-ISL; (ii). not ISL but A-ISL; (iii). neither ISL nor A-ISL.[9] Considering the analyses of Shanghai and Suzhou as a whole, the metrical dominance effect is non-ISL mainly because of the many-to-one tonal mapping. This is a roadblock to traditional linear-based phonology and is the motivation for Autosegmental Phonology in the first place.

On the other hand, having floating tones in the representation does not necessarily make the transformation Non-A-ISL. The Cantonese affixation case is A-ISL (and also ISL) because floating tones in said case are tones without segmental content. However, in Suzhou, floating tones are elements associated with specific morphemes[10] without the TBU-tone association relations. This morpheme-tone association plays a crucial role in the application of tone sandhi, but cannot be expressed under the current model.

4.2 A reanalysis of Suzhou

As I have shown in §3.3, the current A-ISL framework cannot account for the combination effect of left dominance and floating tone representations in Suzhou. For the Suzhou tone sandhi data, the task is to account for the different tonal redistribution status for /HL/ (no redistribution) and /LH/ (even redistribution across two syllables). My floating tone analysis requires the model to recognize tone-morpheme association relations (i.e. which syllable a specific floating tone belongs to underlyingly). One solution is to simply include the tone-

[9](Chandlee and Jardine, 2019) claims that Bounded Meussen’s Rule in Luganda corresponds to the forth possibility — ISL but not A-ISL.

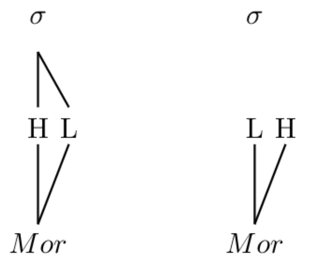
[10]A dominant majority of morphemes in Modern Chinese are monosyllabic. Polysyllabic morphemes are mostly transliterated loanwords. See Lin 2007.

Process	ISL	A-ISL
Cantonese floating H affixation	Yes	Yes
Shanghai left dominant tone sandhi	No	Yes
Suzhou left dominant tone sandhi with floating tones	No	No

Table 1: Summary of analyses

morpheme association information on a separate tier (see a similar treatment with morphological indexes in Trommer and Zimmermann 2014). A refined representational model would look like (24):

(24) a. 'small' b. 'blind'



Shown above are the representations of the two lexical tones in (21). With an additional Mor (morpheme) tier, the representation includes morphological affiliation status of tones by the tone-morpheme association. For instance, both tones in (24b) are floating due to their non-association on the tonal tier. However, the two floating tones are inherent 'members' of the morpheme 'blind' due to their association with Mor on the morpheme tier.

With the added morphological information, the transduction becomes very similar to that of Shanghai. I give the revised model signature and transduction in (25) and (26)

(25) $\langle D|p, s, A, R_{Mor}, P_{Mor}, P_H, P_M, P_L, P_\sigma \rangle$

(26) a. $\sigma'(x) \overset{\text{def}}{=} \sigma(x)$
b. $H'(y) \overset{\text{def}}{=} H(y)$
c. $M'(y) \overset{\text{def}}{=} M(y)$
d. $L'(y) \overset{\text{def}}{=} L(y)$
e. $Mor'(z) \overset{\text{def}}{=} Mor(z)$[11]
f. $A'(x, y) \overset{\text{def}}{=} (A(x, y) \wedge \texttt{first}(x)) \vee (\texttt{first}(x) \wedge \texttt{first}(y)) \vee (\neg A(p(x), y) \wedge R_{Mor}(y, z) \wedge \texttt{first}(z) \wedge \texttt{second}(x) \wedge \texttt{second}(y))$

[11] I remain agnostic regarding the status of morpheme-tone associations (R'_{Mor}) in the output. It is possible that tone-morpheme reassociations also take place through the tone sandhi map. As far as I know, there is no additional post-sandhi phonological process that requires information regarding the morphological association of tones *before* sandhi.

In (25), a unary relation P_{Mor} for morphemes and a binary relation R_{Mor} for tone-morpheme associations are added. In (26), x stands for TBU positions, y for tones and z for morphemes. (26a)-(26e) map all TBUs, tones and morphemes faithfully to the output. (26f) is the revised transduction for the output tone-TBU association relations. A TBU is associated with a tone on the surface in the following three conditions: (i). If the TBU is the first syllable and the tone is associated to itself in the input; (ii). The first tone is by default associated to the first syllable in the output, regardless of its floating status; (iii). If the TBU is the second syllable, and the second tone belonging to the first morpheme (by R_{Mor}) is *not* associated to the first syllable (by A). This transduction ensures that all tonal associations to the first syllable are preserved in the output ((20d), [HL.L]), while a second floating tones of the first syllable can redistribute to the second syllable ((20e), [L.H]). I demonstrate the A-ISL maps for both cases below.

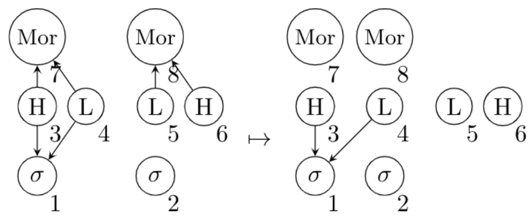


Figure 4: Suzhou A-ISL map — [HL.L]

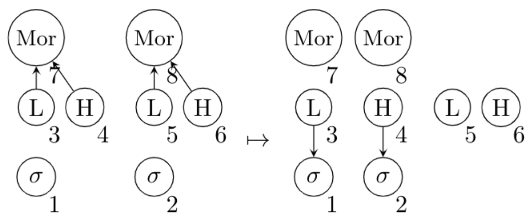


Figure 5: Suzhou A-ISL map — [L.H]

In Figure 4, two tone-TBU association relations $A(3, 1)A(4, 1)$ satisfy the first disjunct of (26f): they are both tone-TBU associations with the first syllable. Morphological information is irrelevant in this case. This gives us the correct output [HL.L]. In Figure 5, however, the latter two disjuncts of (26f) apply: Firstly, the first tone in 3 is associated with the first syllable in 1 in the output; Secondly, the third disjunct of (26f) is satisfied when x is 2, y is 4 and z is 7 — The second tone (position 4) of the first morpheme (position 7) is not associated with the first syllable (position $p(2)$).The output is [L.H], with both tones of the initial morpheme evenly distributed.

5 Conclusion

In this paper, I have examined three distinct tonal processes among Chinese languages assuming an A-ISL framework. I have shown that floating tone suffixation in Cantonese is both ISL and A-ISL, and pure metrical dominance in Shanghai is A-ISL but not ISL. A combination of floating tone representations and metrical dominance in Suzhou tone sandhi is neither ISL nor A-ISL. This is precisely because 'floating tone' has different entailment in the two cases: in Cantonese, a floating tone suffix is simply a tonal element without segmental information; in Suzhou, however, a floating tone is not associated to the TBU underlyingly, but is also part of the lexical representation of specific morphemes. I propose a modified A-ISL model with an additional morpheme tier, and provide a reanalysis of Suzhou by allowing tone-morpheme associations. Crucially, the Suzhou case is problematic for the current A-ISL model *not* due to any model-internal reasons: regardless of the formalism, a working phonological analysis of such pattern has to motivate morphological affiliation independently (Zhu, in prep).

The analyses so far have informed the typology of computational locality and possible Input Strictly Local maps. The three cases I have presented are all definable through Quantifier-Free transductions, and are thus A-ISL. In future work I would like to focus on more cases of linearly ISL but Non-A-ISL tonal maps (as discussed in Chandlee and Jardine 2019), and to see if floating tone representations indeed lead to problems unsolvable under the A-ISL model.

References

Janet Barnes. 1996. Autosegments with three-way contrasts in tuyuca. *International Journal of American Linguistics*, 62(1):31–58.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. dissertation, University of Delaware.

Jane Chandlee. 2018. A computational account of tone sandhi interaction. In *Proceedings of the 2018 Annual Meeting on Phonology*. Linguistic Society of America.

Jane Chandlee and Adam Jardine. 2019. Autosegmental input strictly local functions. In *Transactions of the Association for Computational Linguistics, vol. 7*, pages 157–168. Association for Computational Linguistics.

Jane Chandlee and Steven Lindell. in prep. A logical characterization of strictly local functions. In Jeffery Heinz, editor, *Doing Computational Phonology*. OUP.

Matthew Y. Chen. 2000. *Tone Sandhi: Patterns across Chinese Dialects*. Cambridge University Press, Cambridge.

San Duanmu. 1999. Metrical structure and tone: Evidence from mandarin and shanghai. *Journal of East Asian Linguistics*, 8(1):1–38.

John Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. dissertation, MIT.

Michael Kenstowicz. 1994. *Phonology in Generative Grammar*. Blackwell.

Yen-Hwei Lin. 2007. *The Sounds of Chinese*. Cambridge University Press, New York.

David Odden. 1994. Adjacency parameters in phonology. *Language*, 70(2):289–330.

F. Onn. 1980. *Aspects of Malay Phonology and Morphology*. Ph.D. dissertation, Universiti Kebangsaan-Malaysia, Bangi.

Xinyuan Shi and Ping Jiang. 2013. A prosodic account of tone sandhi patterns in suzhou chinese. In *Proceedings of the 25th North American Conference on Chinese Linguistics*.

Jochen Trommer and Eva Zimmermann. 2014. Generalised mora affixation and quantity-manipulating morphology. *Phonology*, 31(3):463510.

Moira Yip. 2002. *Tone*. Cambridge University Press, New York.

Yuhong Zhu. in prep. A metrical analysis of light-initial tone sandhi in suzhou.

Interpreting Sequence-to-Sequence Models for Russian Inflectional Morphology

David L. King **Andrea D. Sims** **Micha Elsner**
The Ohio State University
{king.2138, sims.120, elsner.14}@osu.edu

Abstract

Morphological inflection, as an engineering task in NLP, has seen a rise in the use of neural sequence-to-sequence models (Kann and Schütze, 2016; Cotterell et al., 2018; Aharoni and Goldberg, 2017). While these outperform traditional systems based on edit rule induction, it is hard to interpret what they are learning in linguistic terms. We propose a new method of analyzing morphological sequence-to-sequence models which groups errors into linguistically meaningful classes, making what the model learns more transparent. As a case study, we analyze a seq2seq model on Russian, finding that semantic and lexically conditioned allomorphy (e.g. inanimate nouns like ZAVOD 'factory' and animates like OTEC 'father' have different, animacy-conditioned accusative forms) are responsible for its relatively low accuracy. Augmenting the model with word embeddings as a proxy for lexical semantics leads to significant improvements in predicted wordform accuracy.

1 Introduction

Neural sequence-to-sequence models excel at learning inflectional paradigms from incomplete input (Table 1 shows an example inflection problem.) These models, originally borrowed from neural machine translation (Bahdanau et al., 2014), read in a series of input tokens (e.g. characters, words) and output, or translate, them as another series. Although these models have become adept at mapping input to output sequences, like all neural models, they are relatively uninterpretable. We present a novel error analysis technique, based on previous systems for learning to inflect which relied on edit rule induction (Durrett and DeNero, 2013). By using this to interpret the output of a neural model, we can group errors into linguistically salient classes such as producing the wrong case form or incorrect inflection class.

Our broader linguistic contribution is to reconnect the inflection task to the descriptive literature on morphological systems. Neural models for inflection are now being applied as cognitive models of human learning in a variety of settings (Malouf, 2017; Silfverberg and Hulden, 2018; Kirov and Cotterell, 2018, and others). They are appealing cognitive models partly because of their high performance on benchmark tasks (Cotterell et al., 2016, and subsq.), and also because they make few assumptions about the morphological system they are trying to model, dispensing with overly restrictive notions of segmentable morphemes and discrete inflection classes. But while these constructs are theoretically troublesome, they are still important for *describing* many commonly-studied languages; without them, it is relatively difficult to discover what a particular model has and has not learned about a morphological system. This is often the key question which prevents us from using a general-purpose neural network system as a cognitive model (Gulordava et al., 2018). Our error analysis allows us to understand more clearly how the sequence-to-sequence model diverges from human behavior, giving us new information about its suitability as a cognitive model of the language learner.

As a case study, we apply our error analysis technique to Russian, one of the lowest-performing languages in SIGMORPHON 2016. We find a large class of errors in which the model incorrectly selects among lexically- or semantically-conditioned allomorphs. Russian has semantically-conditioned allomorphy in nouns and adjectives, and lexically-conditioned allomorphy (inflection classes) in nouns and verbs (Timberlake, 2004); Section 3 gives a brief introduction to the relevant phenomena. While these facts are commonly known to linguists, their importance to modeling the inflection task has not previously

Source	Features	Target
ABAŠ	pos=N, case=NOM, num=SG	ABAŠ
JATAGAN	pos=N, case=INS, num=PL	JATAGANAMI

Table 1: An example inflection problem: the task is to map the Source and Features to the correct, fully inflected Target.

been pointed out. Section 4 shows that these phenomena account for most of Russian's increased difficulty relative to the other languages. In Section 6, we provide lexical-semantic information to the model, decreasing errors due to semantic conditioning of nouns by 64% and of verbs by 88%.

2 Background

The inflection task described above is an instance of the *paradigm cell filling problem* (Ackerman et al., 2009), and models a situation which both computational and human learners face. For humans, the PCFP is closely related to the "wug test" (Berko, 1958): given some previously unseen word, how does a speaker produce a different inflected form? As Lignos and Yang (2016) and Blevins et al. (2017) point out, the same Zipfian distribution that makes other NLP tasks (e.g. MT) difficult is also at play in morphology, namely that no corpus will ever exist that has every wordform from every lexeme. For theoretical morphologists, the difficulty of the PCFP on average is a measure of the learnability of a morphological system, with implications for language typology (Ackerman et al., 2009; Ackerman and Malouf, 2013; Albright, 2002; Bonami and Beniamine, 2016; Sims and Parker, 2016).

Ackerman et al.'s (2009) formulation of the PCFP relies on a simple concatenative model in which words are divided into stems and affixes, and in which each affix is treated as a discrete value. Cotterell et al. (2018) points out that this model is ill-suited to dealing with phenomena like phonological alterations or stem suppletion. Newer models (Silfverberg and Hulden, 2018; Malouf, 2017; Cotterell et al., 2018) use sequence-to-sequence inflection models to avoid these shortcomings.

Faruqui et al. (2016) introduced the use of attention-based neural sequence-to-sequence learning for the inflection task, building on models from machine translation (Bahdanau et al., 2014). Their model treats input as a linear series where grammatical features and characters are encoded as one-hot embeddings and passed to a bidirectional encoder LSTM; output for each paradigm cell is produced by a separate decoder. Kann and Schütze (2016) extended Faruqui et al.'s architecture by using ensembling and by using a single decoder, shared across all output paradigm cells, to account for data sparsity. Later systems (Aharoni and Goldberg, 2017; Kann and Schütze, 2017) have made changes to the input representation and the architecture, for instance incorporating variants of hard attention and autoencoding. From a theoretical standpoint, all these models are "a-morphous" (Anderson, 1992) or "inferential-realizational" (Stump, 2001)— rather than assume a concatenative process which stitches discrete morphemes together into surface word forms, they learn a flexible, generalizable transduction, either between a stem and surface form (Anderson, 1992; Stump, 2001), or between pairs of surface forms (Albright, 2002; Blevins, 2006).

Some older learning-based inflection systems, such as Durrett and DeNero (2013), exploit sequence alignment across strings. Alignment-based systems essentially treat morphology as concatenation. While they do not perform full-scale morphological analysis (since they do not account for phonological alternations), in languages which are mostly concatenative, they do tend to isolate affix-like units as sequences of adjacent insertions or deletions. This property has been criticized in the neural literature (Faruqui et al., 2016) since it represents processes like vowel harmony by enumerating large sets of surface allomorphs, making the learning problem harder. We agree with these criticisms from the modeling standpoint, but we exploit the interpretability of the technique in our analysis of model results.

Our study of Russian concludes that semantically- and lexically-conditioned allomorphy constitutes a problem for current neural reinflection models. This is because such models are trained to map input to output character sequences; they do not typically have access to information about what the words they are inflecting *mean*. We show that, by providing word embeddings as meaning representations, we can reduce this source of error and bring Russian closer to the other languages studied in SIGMORPHON 2016.

Recently the NLP community has also pushed for greater transparency with neural models (xci, 2017; ana, 2019). Wilcox et al. (2018) showed that RNNs learn hierarchical structure in sentences like island constraints. Faruqui et al. demonstrated that RNNs can automatically learn which vowel pairs participate in vowel harmony alternation. Our error analysis allows us to interpret what neural models are learning, reconnecting inflection tasks to linguistic intuitions by generalizing over error classes.

3 Russian Inflectional Morphology

We select Russian as our language of analysis because it was among the three worst-performing languages in the SIGMORPHON 2016 shared task, falling 4+ percentage points behind the other languages. Problems with the design of the Navajo and Maltese datasets may have been the source of the problems with those languages,[1] but this cannot explain the Russian results. The discrepancy hints at some linguistic property which distinguishes Russian from the other languages. Below, we give an overview of the Russian morphological system, concentrating on nouns, verbs, and adjectives, the parts of speech targeted by the SIGMORPHON 2016 shared task.

Russian is an East Slavic language which, in line with other Slavic languages, makes heavy use of inflectional morphology. Russian nouns and verbs belong to inflectional *classes*: groups of words which share a common set of inflectional affixes.

Russian nouns and adjectives have six primary cases—nominative, accusative, genitive, dative, locative, and instrumental—and two numbers, singular and plural. We follow the classification system of Timberlake (2004), which groups nouns into three primary inflection classes (I, II, and III) with subclasses (IA, IB, IIIA, IIIB, and IIIC).

Within these classes, however, the formation of the accusative is further subdivided based on semantics. Specifically, in class IA accusative singular and plural and in classes IB, II, and III accusative plurals, the accusative exhibits syncretism with either the genitive (for animates) or the nominative (for inanimates). In the case of the animate noun STUDENT ('student'), for example, the nominative singular form is *student* and the accusative singular and genitive singular forms are both *studenta*. Conversely, for MESTO ('place'), the accusative singular and nominative singular both have the form *mesto*, but the genitive singular is *mesta*. An example of how this phenomenon looks at the paradigm level for class IA can be seen in Figure 2.

Case	Singular	Plural
Nominative	∅, -', -J, -IJ	-", -I, -II
Accusative	N or G	
Genitive	-A, -JA, -IJA	-OV, -EJ, -EV, -IEV
Dative	-U, -JU, -IJU	-AM, -JAM, -IJAM
Instrumental	-OM, -EM, -IEM	-AMI, -JAMI, -IJAMI
Locative	-E, -II	-AX, -JAX, -IJAX

Table 2: An example of class IA, showing the effect of animacy in the orthography[2] across the singular and plural accusative forms, where *N* or *G* indicate where syncretism occurs in the accusative form based on animacy.

Adjectives in Russian must agree with case, gender, and number of the nouns they modify. They also exhibit the same syncretism in the plural and masculine singular forms, based on the animacy of the noun that the adjective modifies.[3]

Russian also has two verb classes based on what Timberlake calls a verb's *thematic ligature* (i.e. a thematic vowel). A verb is either an *i-conjugation* verb or an *e-conjugation* verb, depending on the vowel used to create the present tense stem. For example, MOLČAT' ('to be silent') forms the present tense stem with *-i* (namely *molč-i-*), making its second person singular form *molčiš'*. Likewise, for a verb like BROSAT' ('to toss'), its present tense stem is *brosae-*, formed with the theme vowel *-e*, making its second person singular form *brosaeš'*. For verbs with monomor-

[1] As announced by the SIGMORPHON shared task organizers.

[2] Examples in this paper are presented in scientific transliteration instead of Cyrillic for accessibility; our system processes Cyrillic characters.

[3] Predicative adjectives have an additional short form which only agrees with gender and number since they only use nominative suffixes. Active participles are inflected as adjectives.

t → č	d → ž	s → š	st → šč
k → č	z → ž	x → š	sk → šč
	g → ž		
p → pl	f → fl	m → ml	
b → bl	v → vl		

Table 3: Russian makes use of phonological alternation, which it encodes orthographically for some characters.

BUMAŽKA → BUMAŽEK (‘paper.DIM’)
NOM.SG → GEN.PL
Gold:

	b	u	m	a	ž		k	a
✓	b	u	m	a	ž	e	k	
						+e		-a

Predicted:

	b	u	m	a	ž		k	a
✗	b	u	m	a	ž	o	k	
						+o		-a

Table 4: Sample induced edit rules can be used to compare gold vs. predicted differences in the MED’s output for error mining. These automatic annotations we subsequently analyzed as missing insertions/deletion and erroneous insertions/deletions.

phemic bases, the class to which the verb belongs (and thus what theme vowel it combines with to form the present tense stem) is not normally thought to be predictable from its syntactic frame or its semantics. It is an idiosyncratic (i.e. lexically-conditioned) property which learners have to memorize for each verb they learn. For verbs with derived bases the situation is more complicated, since derivational suffixes systematically determine the inflection class of a verb. For example, verbs formed with the highly productive *-ova* suffix (*beseda*, ‘conversation’; *besed-ova-t’*, ‘converse’) always belong to the e-conjugation. Transitivity and inflection class are also sometimes related in derived verbs, although not perfectly predictably so. For instance, derived verbs formed with *-i* (e.g. *čist-yj*, ‘clean (adj)’; *čist-i-t’*, ‘clean (verb)’) tend to be transitive (Townsend, 1975).

Verb stems can also undergo phonological alternation, in which the final consonant of a stem changes to another when being inflected for certain parts of the paradigm (e.g. EZDIT’ (‘to ride’) becomes *ezžu* in the first person present singular cell). Further common alternations can be seen in Table 3.

Finally, both nouns and verbs sometimes have morphological stress alternations within the paradigm. These tend to affect high token frequency lexemes, and are thus salient to speakers and learners, but do not affect the majority of words. Counted by type frequency, more than 97% of nouns have fixed stress throughout the paradigm (Brown et al., 2007). Stress alternations are not encoded orthographically.

4 Error Analysis

As mentioned in Section 2, some pre-neural systems for predicting a novel inflected wordform from a source wordform focused on inducing edit operations from one string to another using sequence alignment (Durrett and DeNero, 2013). These approaches model the differences between two strings as a series of *insert* and *delete* operations. While the alignment approach has been superseded by neural models with better performance, we re-apply it here in order to automatically compare and group predicted edit operations vs. gold edit operations. Rather than aligning source to target forms, we align the *gold* target form to the *proposed* target form from the system. For example, if a model learning English plurals incorrectly learned that the ending *-en* was productive, we would see a surplus of *-s* → *-en* errors.

Errors viewed in this way often have natural linguistic interpretations, especially when correlated with the paradigm cells in which they occur. As seen in Table 4, the model correctly predicted the zero genitive plural ending for the noun BUMAŽKA (‘paper.DIM’), but erroneously inserted an *o* (*bumažok*) instead of an *e* (*bumažek*). This is an example of stem alternation in nouns that occurs when there is a zero ending (i.e. nominative singular or genitive plural, depending on the class). The vowel inserted is always an *e* or an *o*, but in this case the wrong vowel was selected.

We used the 2016 SIGMORPHON dataset. Although ideally we would like to have had access to a dataset which more accurately encoded Russian phonology and stress, to our knowledge no such corpus exists. Using the SIGMORPHON dataset, we trained the original MED setup Kann and Schütze made publicly available[4] using the hyperparameters they specified. Other input forms, such

[4] http://cistern.cis.lmu.de/med/

as thos used by Cotterell et al. (2018), are possibly more realistic, but we wished to see why in a controlled setting (i.e. using citation forms) Russian underperformed as compared to languages like Spanish and German. We then extracted errors from the MED system's performance on the validation set, which had 1,591 wordform predictions in total. In using Durrett and DeNero's sequence alignment approach to isolate the differences in edit operations, we were able to annotate each error as a missing deletion (`-d`), an erroneous deletion (`+d`), a missing insertion (`-i`), or an erroneous insertion (`+i`). From here we were able to group erroneous outputs which contained the same edit operations. An example of how we compared and annotated each gold/prediction pair can be seen in Table 5. We can compare these to cases where the same edit operations occur in *correct* answers. This indicates whether an erroneous edit is entirely unattested (i.e. noise), or whether it represents a mis-application of a transform which would have been legitimate for a different source word or target paradigm cell.

We find that the system often produced nouns with the wrong case suffix. In 14% of the total errors, accounting for 29% of all errors affecting nouns, the MED system produced a form of the noun that exists, but corresponds to a different case than the target one. MED also produced verbs with inflections corresponding to the wrong inflection class. These cases account for 10% of the overall errors and 23% of the verb-specific errors. Other errors involved incorrect edits to the stem (in all parts of speech). These accounted for 72% of the overall error rate. These cases were often only a single edit away from the gold wordform, but were more drastic in other cases. We investigated how many of these edits represented mis-applied rules which had been observed elsewhere in training. Surprisingly, *every* erroneous edit rule discovered in the system output had been seen in the training data. We include examples of these error types in Table 7 and summarize the error rates in Table 6.

Many of the noun case errors involve the accusative case, and in particular, an incorrect choice between semantically-conditioned alternatives. As discussed in Section 3, the accusative is syncretic with the genitive or the nominative, conditioned on animacy. In these errors, the system proposes an accusative which matches a correctly inflected form of the word, but not the right one. For instance, the first row of Table 7 shows the proposed accusative of OZNOB 'the chills, shaking'. This matches the genitive form rather than the nominative, which we can easily diagnose by looking for cells in the gold paradigm where the +A edit rule appears.

Verb errors tend to involve alternations characteristic of confusion between i- and e-conjugation verbs. Stem edits often introduce or delete sounds which participate in phonologically motivated alternations, but are not restricted to the contexts in which those alternations legitimately appear.

Error type		Form
Case	✗	OZNOB-A
	✓	OZNOB-∅
	✗	MEXANIZM-OV
	✓	MEXANIZM-Y
Verb class	✗	DOŽD-I-Š'SJA
	✓	DOŽD-E-Š'SJA
Stem edits	✗	REZG-G-OVORČIVY
	✓	RAZ-∅-GOVORČIVY
	✗	ZA-P-O-ŠČ-ENNYJ
	✓	ZA-K-A-Č-ENNYJ
	✗	SANKTPETE-TE-R-B-BUR-B-ŽCAM
	✓	SANKTPETE-∅-R-∅-BUR-∅-ŽCAM

Table 7: Examples of the three main error groups we found produced by the MED system on the 2016 SIGMORPHON dataset. An ✗ is an incorrect prediction and the ✓ below is the gold wordform. Empty set symbols (∅) indicate an erroneous insertion.

5 Model Improvements

In this section, we incorporate a proxy for lexical semantics into the model input representations, leading to improved results. This is useful from a practical standpoint, but also as a clear demonstration that semantic conditioning was responsible for many of the errors which we discussed in the previous section.

As our source for semantic information, we use word embeddings (Mikolov et al., 2013; Socher et al., 2013; Xu et al., 2015). We concatenate the output from the bidirectional encoder with the citation form's embedding. Equipped with this information, the model should be able to learn phenomena like the animacy-dependent syncretism

Gold	Predicted	Rule	Annotation	Category
ABSOLJUTISTA	ABSOLJUŠČISTA	-T+Š+Č	+d+i+i	phonological alternation
DERŽIŠ'SJA	DERŽAEŠ'SJA	-I+A+E	+d+i+i	verb class
ABDOMEN	ABDOMENA	+A	+i	animacy

Table 5: An example of the annotation we performed, where '-' indicates 'missing' and '+' indicates 'erroneous'. Additionally, 'i' indicates 'insertion' and 'd' deletion, so '-i' and '+d' is a missing insertion and erroneous deletion respectively. Collating the grammatical information in the dataset with these annotation allowed us generalize over the errors.

Error type	Percentage	Error Number
Noun class	14%	20
Verb class	10%	15
Stem edits	72%	128

Table 6: A summary of the results from our errors analysis. Results do not sum to 100% since these are only the most frequent errors and can co-occur.

discussed above. We have no *a priori* reason to expect the model to improve its performance on verb class errors, since class membership is a lexical property of the verb stem and not semantically conditioned. However, verbal derivational morphology can affect a verb's meaning and also determines its inflection class, so an indirect effect of semantics is possible. We show below that embeddings are also helpful for verbs, an issue we return to in Section 7.

We modified the original MED code, built in Blocks,[5] so that the output from the encoder could be concatenated with the 300-dimensional word embedding from Kutuzov and Andreev (2015). Since using these embeddings more than doubles the parameter space of the MED system, the model takes longer to converge. We therefore allowed the system to train up to 50 epochs, instead of the 20 Kann and Schütze needed for their models to converge. Both the original MED system and our modified version use early stopping. Once the model has converged, we evaluate system performance by measuring accuracy at the word level.

6 Results

The overall accuracy rates of a single trained MED system and our system are shown in Table 8. Following Kann and Schütze (2016), we also train and evaluate ensembles of five models (Table 9). In each case, our model performs about one percentage point better (significant using McNemar's test). The jump in significance scores between the validation and test is due to the relative sizes of these datasets (1,591 and 22,334, respectively).

System	Val	Test
MED base system	90.03	88.88
MED + word embeddings	91.95*	90.06****

Table 8: Overall results on the validation and test set, using only a single trained model (ensemble of 1). Significance is reported using McNemar's test where * indicates $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, and **** $p < 0.0001$.

System	Val	Test
MED base system	92.14	91.49
MED + word embeddings	93.33*	92.38****

Table 9: Overall results on the validation and test set, using an ensemble of 5 trained models (ensemble of 5). Significance is reported using McNemar's test where * indicates $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, and **** $p < 0.0001$.

We reapply our error analysis to determine error reduction rates by error category. Reductions were largest in noun cases and verb class, with a reduction of more than 50% for both. As seen in Table 10, stem edit errors were least improved. For a breakdown of errors by noun class, see Table 12.

We conduct two other experiments to rule out alternate accounts of the performance increase. First, our model with word embeddings has access to higher-dimensional input for decoding (600 dimensions vs. 300), and therefore to more parameters. We ran a model with 600-dimensional embeddings but no word embeddings, in order to test whether this could be responsible for the gain, but found no significant differences from our baseline system.

Second, we do not expect the word embedding system to encode inflectional information directly (since it operates at the word level with no access

[5] https://github.com/mila-udem/blocks

Error type	Decrease in error rate	Current error rate
Noun class	64.2%	5%
Verb class	88.1%	1%
Stem edits	44.1%	40%

Table 10: Overall error reduction rates in all three error types we considered.

to character information). However, we make absolutely sure that this is not the case by retraining the word embeddings on a stemmed version of our Russian corpus (processed with the NLTK stemmer (Bird, 2006)). Performance using these word embeddings is not significantly different from our results using regular word embeddings.

The error reduction rates by category which we report above are based on the relatively small SIGMORPHON 2016 validation set, and do not represent enough data to conduct statistical analyses by category or paradigm cell. To further break down the improvements quantitatively, we created secondary evaluation sets containing more items. For nouns, we created a secondary evaluation set with the Universal Dependency RusSynTag corpus[6] since it annotates both animacy and gender. We removed any nouns that did not have a 1-to-1 feature correspondence with the SIGMORPHON dataset.[7] This gave us a new evaluation set of 48,590 wordforms. Similarly, we also built a second evaluation set of 25,000 verb forms from Unimorph (Kirov et al., 2016). Although verb conjugation class is not directly annotated, we extracted that information from the second person singular present indicative form. In both cases, we removed any word form that also occurred in the training data.

As seen in Table 11, using word embeddings almost halved the error rate of e-conjugation verbs. It is important to note that the citation form supplied often requires less editing to make an i-conjugation verb than an e-conjugation verb since the citation form often has the *-i* theme vowel. Since the model has a strong preference for reproducing the input, our modification has minimal effect for i-conjugation verbs.

[6] Freely available here: https://github.com/UniversalDependencies/UD_Russian-SynTagRus.git.

[7] These were generally cases where features were missing in the Universal Dependency corpus that were present in the SIGMORPHON corpus.

Verbs	Error count	Total words	Error rate
i-conj	163	516	0.3159
e-conj	430	3191	0.1348
With embeddings			
i-conj	161		0.3120
e-conj	273		0.0856

Table 11: Verb class-specific error reduction rates from 25,000 randomly sampled verb forms from the Unimorph Russian dataset.

Noun class	SG/PL	Error rate	Error rate+	Total count
IA	SG	0.2487	0.2132	2340
	PL	0.4244	0.3839	1555
IB	SG	0.0239	0.0427	1170
	PL	0.1818	0.1439	396
II	SG	0.0542	0.0274	1753
	PL	0.1826	0.1366	805
IIIA	SG	0.0736	0.0851	611
	PL	0.3016	0.1905	126

Table 12: Noun class-specific error reduction rates in the accusative case from 48,590 randomly sampled noun forms from the Universal Dependency RusSynTag dataset. "Error rate+" indicates the error rate after adding word embeddings to the MED system. IIIB and IIIC are not included since there are few nouns and no accusative errors were produced for them by the MED system.

Table 12 shows the general reduction in errors caused by adding word embeddings in various classes of the accusative. We note that errors in accusative forms increase only in class/number combinations that do not exhibit animacy-conditioned syncretism (i.e. singular of classes IB and IIIA).

7 Discussion

What inflectionally useful information is present in the word embeddings? As previously stated, we assume that word embeddings give good clues for noun animacy, but verbs form is not directly conditioned by semantic properties, so we have no *a priori* reason to assume they will indicate verb conjugation. To test whether these features can be derived from the embeddings, we construct maxent classifiers,[8] with only word embeddings as

[8] We use Daumé III (2004)'s implementation available here: http://users.umiacs.umd.edu/~hal/

features, for two binary classification tasks: animate vs. inanimate for nouns and i-conjugation vs. e-conjugation for verbs. Using the same two datasets described in Section 6 for testing nouns and verb class error reduction, we extracted the verb class and animacy annotation along with the citation form's word embedding to create a classification task. With a baseline accuracy rate of 80% for both tasks (i.e. selecting the majority class), both classifiers were more than 98% correct.

We were unsurprised that animacy could be detected in this way, since word embeddings are already used in high-performance models for this kind of lexical feature (Moore et al., 2013; Rubinstein et al., 2015). The model's success for verbs is more surprising. One possible explanation is that Russian verb classes are indirectly related to lexical semantics (Aktionsart). As noted above, derivational suffixes determine the inflection class membership of verbs. Some derivational affixes also create verbs with predictable lexical aspectual properties (e.g. *-nu* creates semelfactives) (Isačenko, 1960; Janda, 2007; Dickey and Janda, 2009), and these semantic properties might be detectable from word embeddings alone. [9] Another possibility is that the predictability of verb class reflects the historical origins of some Russian verbs. Subclasses of verbs borrowed from Church Slavonic tend to have predictable assignments to classes, and also to be more bookish, abstract or metaphorical than native Russian terms (Townsend, 1975; Cubberley, 2002), which may render them recognizable to a distributional system. In any case, the classifier results validate our explanation of why our model improves by showing that the word embeddings do contain the information which the model needs to accurately predict semantically-conditioned allomorphs.

At a higher level, this highlights the issue of semantic conditioning as one which should be taken seriously in models of inflection and the PCFP. Current neural models, which take only word *forms* but not *meanings* as input, are insensitive to this kind of conditioning. They therefore yield overestimates of how difficult it is to acquire and use some morphological systems, such as Russian.

Although our error analysis methods and model extension focused on LMU's 2016 implementation of MED, more recent systems (Aharoni and Goldberg, 2017; Kann and Schütze, 2017) are subject to the same criticisms, since they use the same input representation. In this paper, we focus on Russian, as a language with lower-than-average performance in an inflection task and with a well-described system of inflection classes and alternations. However, we believe it is worth looking for similar effects in less well-studied languages as well, particularly given the wide range of languages now represented in Unimorph (Kirov et al., 2016).

8 Conclusion

Neural networks are a promising technology for cognitive models of a variety of language processing tasks. Their ability to learn flexible representations of complex, multidimensional data allows them to cover a wide range of linguistic phenomena which were difficult to model in more traditional frameworks. In morphology, this corresponds to adopting an "a-morphous" framework in which we do not need to commit to the existence of troublesome constructs like segmentable morphemes. But the adoption of neural nets as cognitive models has demanded a new focus on interpretation. It has become increasingly clear that networks are useful models only to the extent that we can compare what they are learning to what humans learn, and that this is a challenging area of research in its own right.

This work presents a new way to evaluate morphological inflection systems in a linguistically sensitive manner by repurposing previous work in edit rule induction to analyze and group error types. This allows us to attribute errors in inflection generation to specific, interpretable phenomena. We make our code and our expanded datasets publicly available for future use.[10]

We use this new method to discover that semantically- and lexically-conditioned allomorphy are responsible for a shortfall in inflection performance (and thus an overestimate of PCFP complexity) for Russian. Using word embeddings as

`megam/version0_3/`.

[9]Since the data are not tagged for derivational morphology or lexical aspect, it is difficult to assess whether this is a cause of the model's improvement. Given that certain lexical aspects align more naturally with one grammatical aspectual value (perfective or imperfective), we examined whether there is a relationship between verb class and grammatical aspect. We found no correlation in the training or validation data, but this does not rule out the possibility of a lexical semantic effect.

[10]`https://github.com/DavidLKing/SCiL-20`.

a proxy for lexical semantics allows us to supplement the model's input and greatly reduce this source of error. In the future, we will investigate which other languages might show semantically-conditioned allomorphy, potentially even discovering semantic effects in languages where they were not previously known to exist. We will also apply our analysis technique to other models and languages, helping to close the gap between neural reinflection systems and full-scale cognitive models of the PCFP.

9 Acknowledgements

We especially thank Michael White, the Ohio State Computational Linguistics and Language Acquisition discussion groups, Ekaterina Kolbasova, and Katya Rouzina!

References

(2017). *Proceedings of the 1st Workshop on Explainable Computational Intelligence (XCI 2017)*, Dundee, United Kingdom. Association for Computational Linguistics.

(2019). *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy. Association for Computational Linguistics.

Ackerman, F., Blevins, J. P., and Malouf, R. (2009). Parts and wholes: Implicative patterns in inflectional paradigms. *Analogy in grammar: Form and acquisition*, pages 54–82.

Ackerman, F. and Malouf, R. (2013). Morphological organization: The low conditional entropy conjecture. *Language*, pages 429–464.

Aharoni, R. and Goldberg, Y. (2017). Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2004–2015.

Albright, A. C. (2002). *The identification of bases in morphological paradigms*. PhD thesis, University of California, Los Angeles.

Anderson, S. R. (1992). *A-morphous morphology*, volume 62. Cambridge University Press.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Berko, J. (1958). The child's learning of English morphology. *Word*, 14(2-3):150–177.

Bird, S. (2006). Nltk: The natural language toolkit. In *COLING ACL 2006*, page 69.

Blevins, J. P. (2006). Word-based morphology. *Journal of Linguistics*, 42:531–573.

Blevins, J. P., Milin, P., and Ramscar, M. (2017). The Zipfian paradigm cell filling problem. *Perspectives on morphological organization: Data and analyses*, 10:141.

Bonami, O. and Beniamine, S. (2016). Joint predictiveness in inflectional paradigms. *Word Structure*, 9(2):156–182.

Brown, D., Corbett, G. G., Fraser, N., Hippisley, A., and Timberlake, A. (2007). Russian noun stress and Network Morphology. *Journal of Linguistics*, 34:53–107.

Cotterell, R., Kirov, C., Hulden, M., and Eisner, J. (2018). On the complexity and typology of inflectional morphological systems. *arXiv preprint arXiv:1807.02747*.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., and Hulden, M. (2016). The SIGMORPHON 2016 Shared Task—Morphological Reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Cubberley, P. (2002). *Russian: A linguistic introduction*. Cambridge University Press.

Daumé III, H. (2004). Notes on CG and LM-BFGS optimization of logistic regression. Paper available at http://pub.hal3.name#daume04cg-bfgs, implementation available at http://hal3.name/megam/.

Dickey, S. M. and Janda, L. (2009). *Xoxotnul, sxitril*: The relationship between semelfactives formed with *-nu-* and *s-* in Russian. *Russian Linguistics*, 33:229–248.

Durrett, G. and DeNero, J. (2013). Supervised learning of complete morphological paradigms. In *HLT-NAACL*, pages 1185–1195.

Faruqui, M., Tsvetkov, Y., Neubig, G., and Dyer, C. (2016). Morphological inflection generation using character sequence to sequence learning. In *Proc. of NAACL*.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Isačenko, A. (1960). *Grammatičeskij stroj russkogo jazyka v sopostavlenii s slovackim – Čast' vtoraja: morfologija*. Slovackoj Akademii Nauk.

Janda, L. (2007). Aspectual clusters of Russian verbs. *Studies in Language*, 31:607–648.

Kann, K. and Schütze, H. (2016). MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. *ACL 2016*, page 62.

Kann, K. and Schütze, H. (2017). The LMU system for the CoNLL-SIGMORPHON 2017 shared task on universal morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 40–48. Association for Computational Linguistics.

Kirov, C. and Cotterell, R. (2018). Recurrent neural networks in linguistic theory: Revisiting Pinker and Prince (1988) and the past tense debate. *arXiv preprint arXiv:1807.04783*.

Kirov, C., Sylak-Glassman, J., Que, R., and Yarowsky, D. (2016). Very-large scale parsing and normalization of wiktionary morphological paradigms. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Kutuzov, A. and Andreev, I. (2015). Texts in, meaning out: neural language models in semantic similarity task for russian. *Dialog 2015*.

Lignos, C. and Yang, C. (2016). *Morphology and Language Acquisition*, page 743764. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Malouf, R. (2017). Abstractive morphological learning with a recurrent neural network. *Morphology*, 27(4):431–458.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Moore, J., Burges, C. J., Renshaw, E., and Yih, W.-t. (2013). Animacy detection with voting models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 55–60.

Rubinstein, D., Levi, E., Schwartz, R., and Rappoport, A. (2015). How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 726–730.

Silfverberg, M. and Hulden, M. (2018). An encoder-decoder approach to the paradigm cell filling problem. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2889.

Sims, A. D. and Parker, J. (2016). How inflection class systems work: On the informativity of implicative structure. *Word Structure*, 9(2):215–239.

Socher, R., Bauer, J., Manning, C. D., et al. (2013). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465.

Stump, G. T. (2001). *Inflectional morphology: A theory of paradigm structure*. Cambridge University Press.

Timberlake, A. (2004). *A reference grammar of Russian*. Cambridge University Press.

Townsend, C. E. (1975). *Russian word-formation*. Slavica Publishers.

Wilcox, E., Levy, R., Morita, T., and Futrell, R. (2018). What do RNN language models learn about filler–gap dependencies? *EMNLP 2018*, page 211.

Xu, W., Auli, M., and Clark, S. (2015). CCG supertagging with a recurrent neural network. In *ACL (2)*, pages 250–255.

Stop the Morphological Cycle, I Want to Get Off: Modeling the Development of Fusion

Micha Elsner, Martha Booker Johnson, Stephanie Antetomaso, Andrea D. Sims
Department of Linguistics
The Ohio State University
elsner.14,johnson.6713,antetomaso.2,sims.120@osu.edu

Abstract

Historical linguists observe that many fusional (unsegmentable) morphological structures developed from agglutinative (segmentable) predecessors. Such changes may result when learners fail to acquire a phonological alternation, and instead, "chunk" the altered versions of morphemes and memorize them as underlying representations. We present a Bayesian model of this process, which learns which morphosyntactic properties are chunked together, what their underlying representations are, and what phonological processes apply to them. In simulations using artificial data, we provide quantitative support to two claims about agglutinative and fusional structures: that variably-realized morphological markers discourage fusion from developing, but that stress-based vowel reduction encourages it.

1 Introduction

While modern typologists reject the wholesale categorization of languages as isolating, agglutinative or fusional (Haspelmath, 2009), they still recognize a distinction between morphological structures which can be easily segmented and those which cannot (Plank, 1999). In ones with morphological *fusion* (or *cumulation*), multiple morphosyntactic properties (MSPs)[1] are realized by a single morph with no immediately segmentable pieces.[2] For instance, Turkish *tarla-lar-ı* and Old English *feld-a* both indicate 'field-PL.ACC' (Plank, 1999), but the Old English suffix cannot be further analyzed whereas the Turkish word has separate number and case morphemes.

Along with this taxonomic distinction comes a historical origin story, sometimes called the *morphological cycle* (Hock and Joseph, 1996)[183]. Through processes of phonological reduction, independent function words become attached to content words as agglutinative inflections. Further phonological reduction or sound changes blur the boundaries between morphemes, leading to fusion. Finally, affixes may become so non-transparent that their association with MSPs is lost (demorphologization) at which point new function words may be recruited to replace them, beginning the cycle anew.

Morphological change is more various and more complicated than this simple story suggests, and this cycle isn't the only way in which fusion can arise (Grünthal, 2007; Igartua, 2015; Karim, 2019). However, it is one way that has been observed. In this paper we focus on the role of phonological processes in the transition between agglutination and fusion. Morphological reanalysis often results from an interaction between the phonology of a language and the learning mechanism. Specifically in this context, morphemes are most likely to fuse if the environments in which they occur, and the phonological processes triggered by those environments, are vulnerable to reanalysis, which is to say, to mis-learning. The question becomes: which kinds of phonological processes are likely to make morphological constructions vulnerable to reanalysis, and which are not?

In order to test the role that phonological processes play in making agglutinative structures vulnerable to reanalysis, we provide a formal learning model[3] for morphological systems whose internal representations clearly distinguish between agglutination and fusion. The model extends Cot-

[1] We use *morphosyntactic category* to refer to sets of properties; cross-linguistically common categories are TENSE, PERSON, NUMBER, etc., and morphosyntactic properties are PRESENT, PAST, etc.

[2] Following practice in morphology, we use the term *morph* to refer to (only) the form part of a morpheme.

[3] Code and data at github.com/melsner/scil2019-fusion.

terell et al. (2015), learning a Bayesian model which maps from sets of MSPs to surface forms in three steps: selection of a morphological template, concatenation of underlying forms, and phonology. We validate the model by testing on a series of artificial languages. The model recovers the expected analyses for prototypically agglutinative or fusional languages; for languages which can be analyzed in either way, we demonstrate in the first study that those with variably-realized morphological markers (i.e. ones that are sometimes present, sometimes absent) are less likely to be learned as fusional. In a second study, we show that languages with stress-based vowel reduction are more likely to be learned as fusional. Our model thus provides quantitative support for previous observations that languages with large proportions of agglutinative structures also frequently have large numbers of variably-realized morphs (Plank, 1999) and vowel harmony rather than stress-based reduction (Zingler, 2018).

2 Related work

	Indo-European		Ancient Greek	
	PRS	AOR	PRS	AOR
1SG	*-m-i	*-m	dídō-mi	édō-n
2SG	*-s-i	*-s	dídō-s	édō-s
3SG	*-t-i	*-t	dídō-si	édō

Table 1: Partial set of Indo-European and Ancient Greek ('give') person-number forms in present indicative and aorist

We begin with a concrete example of the kind of morphological change we are describing. In some Indo-European (IE) athematic verbs, person and number were expressed cumulatively but tense was realized via a separate morpheme: *-i* for present active indicative and zero for aorist active indicative (Table 1). (These endings are reconstructed for IE but attested in Sanskrit.) However, sound changes between IE and Proto-Greek obscured the unity of the person-number morphs across present and aorist. For example, word-final [m] turned into [n] as a result of sound change, resulting in different 1SG forms in Ancient Greek.[4] These changes led speakers to reanalyze the formerly separate morphemes as fused (Brian Joseph, p.c.): 1SG.PRS *-mi* vs. 1SG.AOR *-n*. This reanalysis is evidenced by the fact that in Aeolic dialects, speakers extended the athematic ending *-mi* to verbs that did not historically have it, giving, e.g., *fíli-mi* 'love-1SG.PRS' where *filõ* is expected etymologically. The fact that *-mi* was extended as a single unit indicates that it had undergone fusion.

The reanalysis of the Greek suffixes was thus driven by sound changes that introduced phonological alternations, and in the process introduced ambiguity regarding the morphological structure. In the wake of these changes, speakers were faced with an analytic choice, e.g.: is there one 1SG morpheme *-m* plus a phonological rule, or different 1SG endings *-mi* and *-n* that also express tense?

The extent to which sound change leads agglutinative structures to be reanalyzed as fusional has recently been questioned (Haspelmath, 2018).[5] Nonetheless, this kind of ambiguity between analyses at different levels of representation is often a driver of language change (Bybee, 1999) and phonological reduction of agglutinative structures is widely cited as a source of fusionality (Bybee, 1997; Igartua, 2015, among others). Just as phonological rules and categories can arise when low-level phonetic processes like assimilation are reanalyzed as phonological, so fusion can appear when the effects of phonological process are "baked in" to the morphological representations. Bybee (2002) summarizes the idea (with reference mostly to syntax) with the catchphrase: "Items that are used together fuse together."

Both Heath (1998) and Zingler (2018) point out the implication that agglutinative constructions must have "barriers"— typological features which prevent them from becoming fusional.[6] Zingler makes a specific proposal, that fixed (lexical) stress systems tend to encourage fusion, while vowel harmony discourages it. This builds on a typological observation: the kinds of phonological alternations that occur in agglutinative and fusional systems tend to differ, "... with vowel harmony tending to imply agglutination" (Plank, 1999)[310].[7] Zingler argues that fixed stress leads

[4]Also, prior to Proto-Greek [t] deleted in some contexts, affecting the 3SG.AOR, and between Proto-Greek and attested Greek $[t] \rightarrow [s]$ (Brian Joseph, p.c.). Both were regular sound changes but had consequences for morphology.

[5]In fact, Haspelmath states (pp107-8) that "...we do not know how it is that robust inflectional patterns with cumulative and suppletive affixes arise". Our paper offers a partial answer.

[6]The argument of Heath (1998) applies to the first (isolating-agglutinative) step of the cycle, rather than the second (agglutinative-fused) as discussed here: he suggests that established agglutinative systems grammaticalize independent function words into morphemes more quickly, due to their analogical similarity to existing morphemes.

[7]An anonymous reviewer questioned the basis for this

to reduction in unstressed syllables, which over time may lose their vowels, placing their consonants in new environments with varied phonological effects. Harmony, on the other hand, prevents the loss of vowels, while at the same time indicating that bound elements are part of the phonological word (since they undergo harmonic changes based on the word stem).[8]

One question here has to do with the relationship between language-level and construction-level properties. From Haspelmath's perspective, individual constructions may be agglutinative or fusional, but it is not clear that languages as a whole fall into cleanly defined types. However, Zingler's proposal is rooted in phonology-morphology interactions. Phonological processes generally operate across a range of constructions in a language. Phonological properties are language-level and thus might be expected to have an across-the-board effect on morphological structure. Moreover, accumulation of effects on individual constructions may result in a disproportionate number of constructions of the same type (agglutinate, fusional, etc.) in a given language. In other words, there is no expectation that the ways constructions develop historically will be fully independent of each other. To the extent that the phonological context is the same for different morphological constructions, we might expect similar pressures in and outcomes of language change. While Zingler himself does not say so, his ideas stand as an implicit challenge to Haspelmath's questioning of the validity of morphological types at the language level.

We argue below that the presence of variably-realized morphological marking is also a protective factor against fusion. Many agglutinative languages have position classes that are sometimes filled by an overt morph, and sometimes not; this is what we mean by 'variably-realized' morphological marking. Examples include morphosemantic markers such as causatives, desideratives or negatives, whose position class slots are filled only when that meaning occurs, and optional agreement marking (Plank, 1999). Polysynthetic languages, which are invariably mostly agglutinative, contain even more variably-realized elements, such as incorporated objects (Comrie, 1989). We suggest that, because variably-realized elements break up sequences of morphemes that would otherwise always appear next to one another, they render fusional analyses less appealing to the learner. Our argument not only explains the previous observation that variably-realized marking and agglutination correlate, but might also help to explain where and how fusionality develops.

1	Input MSPs	M_1=I, M_3=I, STEM=1
	Transducer 1: fusion	
2	Abstract ms	M_1=I$|$$M_3$=I, STEM=1
	Transducer 2: lexicon	
3	Underlying	mwi-mela
	Transducer 3: phonology	
4	Surface form	mwimela

Figure 1: Overall architecture of our model, consisting of three finite-state transducers, producing two intermediate layers of latent representation (in gray).

Caballero and Kapatsinski (to appear) quantify fusionality in the Uto-Aztecan polysynthetic language Choguita Rarámuri. They show that morphemes exhibit some fusion, especially close to the stem. Their research focus is similar to ours in examining how learners might infer morphological boundaries. However, their approach differs from our own in two ways. First, it provides a description of how much fusion is present based on the Naive Discriminative Learner (Baayen et al., 2011) and some variant models, but not a causal model of how language properties encourage or discourage fusion. Second, it lacks an explicit model of phonological rules. Caballero and Kapatsinski point out that if learners can mentally "undo" the effects of regular phonological rules, the Naive Learner will overestimate the degree of fusionality. The model we present below is designed to test causal mechanisms underlying the development of fusionality, and specifically the role of phonological rules.

generalization, pointing out that both Algonquian and Nilotic languages have vowel harmony, but the former would be classified as agglutinative and the latter as fusional. While we agree with the reviewer that more typological investigation is warranted, we follow Plank and others in the claim that there is a typological correlation to be explained.

[8] Plank (1998)[201] points out that this idea of vowel harmony 'cementing' the internal cohesion of agglutinative word structure goes back to Baudouin de Courtenay (1876), but is not unproblematic in its reasoning. In our work, nothing depends on vowel harmony creating greater word-internal cohesion.

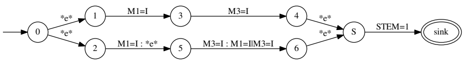


Figure 2: Fragment of first transducer, from MSPs to abstract morphemes. (For compactness, only one MSP per morphosyntactic category is shown.)

3 Model

Our model is intended to capture the first stage of the transition from agglutination to fusion, in which the learner reanalyzes an ambiguous polymorphemic structure as monomorphemic. This reanalysis is covert, affecting only the learner's mental representation; in order for the surface system to become unambiguously fusional (i.e. for the change to become actualized, in the terminology of historical linguistics), the reanalyzed marker must generalize to other words, as we saw above for Greek, or undergo further diachronic changes. We leave modeling such changes for future work.

The model (Figure 1) formalizes our intuitions about agglutinative and fusional analyses of morphological systems. In order to do so, it represents morphemes as invariant underlying representations and applies phonological processes that transform them into surface forms. Because the popular sequence-to-sequence framework for inflection (Kann and Schütze, 2016) conflates these processes within a single neural network, we choose instead to extend an older model, Cotterell et al. (2015), in which these components are separate. While this model may be less capable overall, it is more interpretable in terms of the theoretical questions we are trying to answer.

Cotterell et al. model the correspondence between sequences of abstract morphemes and surface strings. The term "abstract morpheme" refers to a set of MSPs that already reflect the effects of fusion— in the context of agglutination, each abstract morpheme is a single MSP, whereas for fusion, the abstract morphemes bundle together many MSPs. The model maps abstract morphemes to surface strings in the following steps: first, each abstract morpheme is assigned an underlying phonological form; next, these forms are concatenated to yield an underlying inflected form; finally, this form is passed through a finite-state transducer which applies (stochastic) phonological rules. (Lines 2-4 of Figure 1.)

Our model differs from theirs primarily in adding a new initial step, which maps a sequence of atomic MSPs into a corresponding sequence of abstract morphemes. This is the step at which fusion occurs. For instance, a sequence *STEM=give,NUM=PL,TENSE=PRS* could be output as three separate symbols, or as *STEM=give*, *NUM=PL|TENSE=PRS*, where we use the | notation to indicate that two MSPs are fused into a single abstract morpheme. The model simplifies slightly by requiring uniformity at the level of morphosyntactic categories; in our illustrating example, either all combinations of number and tense MSPs would be fused or none would be.[9]

For simplicity, we also modify the model so that it consists of a cascade of relatively small finite-state transducers (FSTs) (Mohri et al., 2002) which we can implement using the Carmel package (Graehl, 1997). This necessitates some changes and simplifications to the model, but allows us to use Carmel's built-in Bayesian inference (Chiang et al., 2010) rather than belief propagation as in Cotterell et al. (2015).

As stated, the first transducer in the cascade maps a sequence of MSPs into a sequence of abstract morphemes (without specifying any phonological detail). For computational convenience, we make two simplifying assumptions: The input MSPs are provided in a fixed, templatic order (Stump, 1997), in which only contiguous subsequences can be fused. MSPs are not allowed to fuse with the stem (that is, there is no MSP-conditioned stem allomorphy), even though this occurs in real languages. The transducer (Fig 2) first chooses an allowable fusion template via epsilon transition and then deterministically transforms the input sequence.

The second transducer is a lexicon (Figure 3) which maps each abstract morpheme to a phonological underlying form. Cotterell et al. implement this as a distribution of point masses on strings, which is intractable and must be approximated.[10] We use a simpler solution which is finite-

[9]In real languages, individual MSPs (or even individual allomorphs of MSPs) can fuse, even when other MSPs belonging to the same categories do not. Stump (2001)[139–144] gives examples under the heading of 'portmanteau rule blocks'. In Swahili verbs, subject agreement prefixes and the negative prefix *ha-* normally have separate realizations and occupy adjacent position classes. However, the combination of 1SG.SBJ (normally *ni-*) and NEG is realized as a single, fused prefix *si-*.

[10]It has the advantages that strings are not limited in length, and that the morpheme may vary over two unrelated phono-

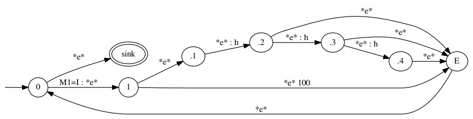


Figure 3: Fragment of second transducer, from abstract morphemes to characters. Only the lexical entry for M_1=1, only three steps in the linear chain, and only the character h are shown.

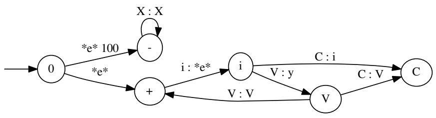


Figure 4: Fragment of third transducer responsible for altering i to y. X stands for any character, V for any vowel and C for any consonant.

state and tractable. Each word in the lexicon has an initial state with two outgoing epsilon transitions; one leading back to the start state (thus producing a null morpheme) and another leading to a linear chain of 15 states. Each state in the chain can produce any non-null character, or transition back to the start. This transducer can produce any string up to 15 characters long; the posterior tends to concentrate around a single underlying form per morpheme. We set the prior odds ratio for the two initial transitions so that the null morpheme is 100 times more likely *a priori* than the linear chain. This prior biases the model toward parsimonious analyses with smaller morpheme inventories, provided they can satisfactorily account for the data.

The third transducer (Figure 4) implements phonological rules. While Cotterell et al. supply a full finite-state phonology (Riggle, 2004; Hayes and Wilson, 2008, and others), in our experiments below, we use a custom machine implementing only the specific rules which actually exist in our artificial language. However, the machine executes the rules non-deterministically; the system must learn the true probability with which the rules occur. Again, we use prior parameters to determine how much evidence is necessary to convince the system that a phonological rule is justified. In our experiments below, we set the prior odds ratio of the rule applying to 1:100. In simulation $\mathcal{C}$, we vary the strength of the prior (by multiplying the prior counts by a constant α) and report

logical forms without reserving mass for "hybrid" versions.

Underlying	Surface	Gloss
ndi-i-ko:mala	ndi:ko:mala	'I am sitting'
u-i-ko:mala	wi:ko:mala	'You.SG are sitting'
a-i-ko:mala	i:ko:mala	'S/he is sitting'
tu-i-ko:mala	twi:ko:mala	'We are sitting'
mu-i-ko:mala	mwi:ko:mala	'You.PL are sitting'
va-i-ko:mala	vi:ko:mala	'They are sitting'

Table 2: Conjugation of a Kihehe verb in the present tense (Johnson, 2015).

results as a function of this parameter.

We perform posterior inference using blocked Gibbs sampling (Chiang et al., 2010). For each language, we run 20 Markov chains with random starting points, annealing linearly from temperature 4 to 1 over 200 iterations. We average the final counts from each chain to obtain the posterior.

4 Case study 1: Variably-realized marking

In this section, we run a series of simulations on artificial languages, intended to be reminiscent of the Bantu language Kihehe, spoken in Tanzania (Lewis, 2009). Simulations $\mathcal{A} - \mathcal{B}$ show that the model can learn both agglutinative and fusional systems; $\mathcal{C}$ shows that the model's preference for fusionality is dependent on the phonological prior weight α. $\mathcal{D}$ gives the main conclusion, that the presence of a variably-realized marker between two obligatory ones can block the emergence of fusion.

We first give a brief overview of Kihehe itself. Kihehe verbs are marked for person-number agreement with the subject; the form of the agreement marker reflects the noun class of the subject. This marker is sometimes followed by a tense marker. Although Kihehe has morphemes which begin with vowels, its phonological rules act to prevent onsetless syllables from surfacing, by transforming the first vowel in a VV sequence into a glide, or deleting one vowel, and in both cases, lengthening the remaining vowel (Odden and Odden, 1999).[11] This creates a system in which agreement and tense markers are arguably fused on the surface (Table 2). In 3SG and 3PL, where vowel deletion occurs, segmentation is impossible. In the other cells, segmentation of the surface form is possible but gliding prevents postulation of a single, invariant form of each agree-

[11] We present these phonological processes here as SPE rules, although of course other theoretical frameworks like OT could derive the same results.

Name	M_1	M_2	M_3	Phonology	Examples
$\mathcal{A}$	$\begin{cases} ta \\ ko \\ he \\ mu \\ gu \\ si \end{cases}$	-	$\begin{cases} i \\ a \\ de \\ no \end{cases}$		koimela, muimela
$\mathcal{B}$	$\begin{cases} ya, se, dunu, lanu \\ ha, hi, si, yu \\ yi, wa, bise, logi \\ \ldots \end{cases}$				dunumela, yamela
$\mathcal{C}$	as $\mathcal{A}$	-	as $\mathcal{A}$	$\begin{bmatrix} \text{V} \\ \text{+high} \end{bmatrix} \rightarrow$ glide / _V V $\rightarrow \phi$ / _V	mwimela (< mu-i-mela), kamela (< ko-a-mela)
$\mathcal{D}$	as $\mathcal{A}$	$\begin{cases} sa \\ \epsilon \end{cases}$	as $\mathcal{A}$	as $\mathcal{C}$	mwimela (< mu-i-mela), musimela (< mu-sa-i-mela)

Table 3: Morphophonology of four simulated languages (case study 1).

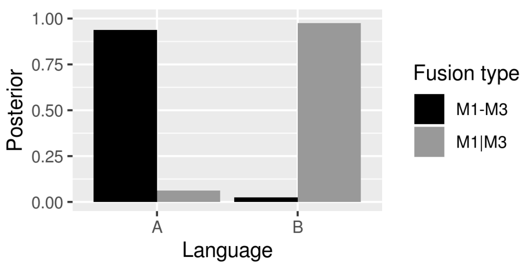


Figure 5: Probability of fusion in $\mathcal{A}$ vs $\mathcal{B}$.

ment marker. This parallels the conditions in pre-Greek that led to reanalysis of separate person-number and tense morphemes as fused (Table 1 above).

As in Kihehe, our artificial languages have stems made up of CV syllables. We use an inventory of 5 vowels and 15 consonants; for each language, we generate 200 unique random stems, with length $\min(5, Geom(.5))$, which we use to create a corpus of 1000 inflected forms. Each language has two required morphosyntactic categories, M_1 and M_3 (e.g., person and tense), realized as prefixes, with uniformly distributed values (MSPs). In simulation $\mathcal{D}$, we explore the impact of a variably-realized category M_2 which appears between the two. Table 3 shows the realizations of M_1, M_2 and M_3 and the phonology in each simulation.

Language $\mathcal{A}$ is prototypically agglutinative. Each category:property (MSP) pair licenses a unique, segmentable morph in the surface string. (The morphs that realize M_1 contain equal numbers of high and low vowels, and for M_3 contain equal numbers of vocalic and consonantal onsets.) Language $\mathcal{B}$ is prototypically fusional. While words inflect for the same categories as in language $\mathcal{A}$, each M_1, M_2 value pair licenses a unique morph that realizes both categories (a sampled string of one or two syllables). We expect the model to analyze $\mathcal{A}$ as agglutinative, due to the prior preference for a small morpheme inventory (the agglutinative analysis has 6+4=10 morphemes while the fusional analysis has 6*4=24), and $\mathcal{B}$ as fusional; this is the actual result (Figure 5).

Language $\mathcal{C}$ has the same underlying properties as language $\mathcal{A}$, but is subject to phonological rules which result in non-isomorphic relationships between form and meaning in the surface forms. (The surface prefixes are thus segmentable, but not into invariant forms; for example, *ko-* alternates with *k-* and *mu-* with *mw-*, conditioned on their phonological environment.) We use language $\mathcal{C}$ to explore the effects of the prior parameter α, which encodes our bias against using the phonological rule; larger α means that more evidence is required to justify the rule's existence. Not surprisingly, small α leads to agglutinative analyses, while large α leads to fusion (Figure 6, top).

Finally, we investigate the effects of M_2, a variably-realized category between M_1 and M_3, using language $\mathcal{D}$. For this simulation, we set $\alpha = 1000$, a setting which we found in the previous experiment would result in a fusional analysis. We do so because we are interested in whether M_2 can *prevent* fusion from occurring; thus, it makes sense to start from a setting in which fusion is expected. All versions of language $\mathcal{D}$ have the category M_2 between M_1 and M_3, but we vary the

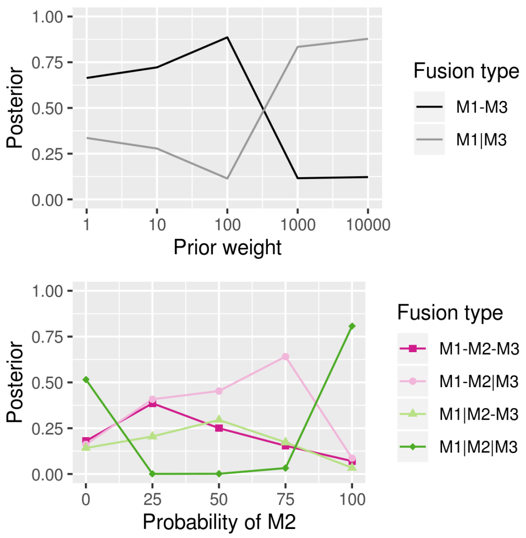


Figure 6: Fusion in (top) $\mathcal{C}$ as a function of α, (bottom) $\mathcal{D}$ as a function of the probability of non-zero M_2.

probability with which it takes its non-zero value (realized as *sa-*). We find (Figure 6, bottom) that when *sa-* always or never occurs, the posterior mode is a fully fused system, $M_1|M_2|M_3$. But when *sa-* is variably realized, full fusion essentially never occurs. Instead, we find either agglutination (M_1-M_2-M_3, the plurality outcome when $p(\text{sa}) = .25$) or partial fusion, in which M_2 is realized jointly with one of its neighbors.

Thus, the important result is that in the context of phonological rules that create surface-ambiguous word-forms, variably-realized morphemes decrease the likelihood of agglutinative morphemes being reanalyzed as fusional.

5 Case study 2: Stress-based vowel reduction

Our next study addresses Zingler's claims about Turkish agglutination. Zingler argues (p422) that languages have various mechanisms for articulatory reduction of vowels. One of these is vowel harmony, which replaces some distinctive features of a vowel with those of its neighbor, and another is durational reduction, which reduces a vowel's absolute length, and tends to erode its features by centralizing it. These mechanisms are complementary; harmony correlates with syllable-timed languages and with systems that assign stress to a fixed syllable relative to the word boundary. Durational reduction correlates with stress-timed languages and with systems in which the stressed syllable is lexically determined. Zingler's hypothesis is that durational reduction leads to fusion, and that vowel harmony, as an alternative way to ease articulation without durational reduction, is what prevents Turkish from becoming fusional.[12]

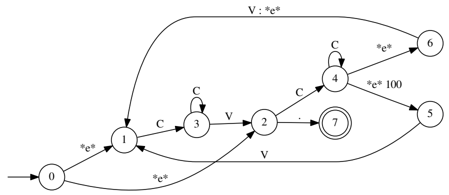


Figure 7: The transducer for vowel reduction (with final stress).

In this section, we validate Zingler's claim that vowel reduction tends to encourage the development of fusion, and add to his idea by showing that this is especially so when the position of reduction is predictable within particular morphemes. Simulation $\mathcal{E}$ investigates the case where stress is predictable within morphemes, and $\mathcal{F}$ the case where it is not. As above, we use artificial languages in which stems consist of CV syllables. Languages in this section have two required categories, M_1 and M_2, realized as suffixes. Table 4 shows the realizations of M_1 and M_2 and the phonology.

We next apply vowel reduction. In simulation $\mathcal{E}$ we apply final stress and then alternate strong and weak syllables moving left; the reduction rule deletes each weak vowel with some probability.[13] So, the word *dite-ko-**de*** in fully reduced form would become *dtekde*. Simulation $\mathcal{F}$ is similar, but with initial stress, so *dite-k**o**-de* would become *ditkod*. Such stress rules follow from the core predictions of metrical stress theory (Hayes, 1995).[14] Within each simulation, we compare languages with varying rates of reduction, ranging from no reduction to all unstressed vowels reduced.

Although neither $\mathcal{E}$ nor $\mathcal{F}$ has a true lexical stress system, the varying stress rules have implications for the predictability of stress placement

[12]Zingler also argues that vowel harmony helps maintain a morpheme minimality criterion. He does not consider whether morpheme minimality plays a role in preventing fusion, but we believe this could also be relevant and could be simulated in our model, with suitable alterations to the lexicon. But we leave doing so for future work.

[13]This approximates the 'fall of the jers', a sound change in the history of the Slavic languages (Kiparsky, 1979).

[14]Kager (1995) gives example languages which have the stress systems described here. Weri parses feet from right-to-left, with final stress; Hungarian parses from left to right with initial stress.

Name	M_1	M_2	Phonology	Examples
$\mathcal{E}$	$\begin{cases} ta \\ ko \\ he \\ mu \\ gu \\ si \end{cases}$	$\begin{cases} pi \\ ka \\ de \\ no \end{cases}$	Assign sS stress from right $[\ \text{C voice-}x\ ] \rightarrow [\ \text{voice-}y\ ]\ /\ \_[\ \text{C voice-}y\ ]$	ddekte (< dite-ko-de)
$\mathcal{F}$	As $\mathcal{E}$	As $\mathcal{E}$	Assign Ss stress from left $[\ \text{C voice-}x\ ] \rightarrow [\ \text{voice-}y\ ]\ /\ \_[\ \text{C voice-}y\ ]$	ditkod (< dite-ko-de)

Table 4: Morphophonology of two simulated languages (case study 2).

on morphemes. Because each word has two obligatory suffixes, final stress (simulation $\mathcal{E}$) means that the second suffix, corresponding to M_2, will always be pronounced with a full vowel, while the suffix for M_1 will be probabilistically reduced. The same condition would hold in a true lexical stress language, although in such a language it would *also* hold if the number of suffixes were variable. In $\mathcal{F}$, however, stress lands on the suffix realizing M_1 when the length of the stem is even, on the suffix for M_2 when it is odd. Thus, each suffix appears in both strong and weak positions.

Vowel reduction disrupts the original CV structure of our languages, allowing consonant clusters to appear on the surface. It is extremely common for such clusters to simplify for articulatory reasons (Brohan and Mielke, 2018)— we apply only one simplification rule, progressive voicing assimilation. Thus, *dtekde* would surface as *ddekte*. In a real language, we might expect further simplifications to apply to prevent, for instance, geminate *dd* at the beginning of a word; for our purposes, however, a single assimilation rule is sufficient.

We apply the same learning procedure as in the previous section. The feature and lexicon transducers are unchanged. The transducer for vowel reduction is shown as Figure 7; the transducer for assimilation resembles the one in Figure 4. We use $\alpha = 1000$ as a bias parameter to penalize both phonological rules (vowel reduction and consonant assimilation).

Figure 8 (top) shows the results for language $\mathcal{E}$. With reduction rate 0 (no reduction), the posterior mode is an agglutinative system. Optional vowel reduction (25-75%) produces mixed systems in which both agglutination and fusion are recognized as possible analyses, although the posterior probability of fusional analyses climbs slightly as reduction increases. With 100% reduction, the posterior strongly prefers fusion.

The orange line shows the posterior probability

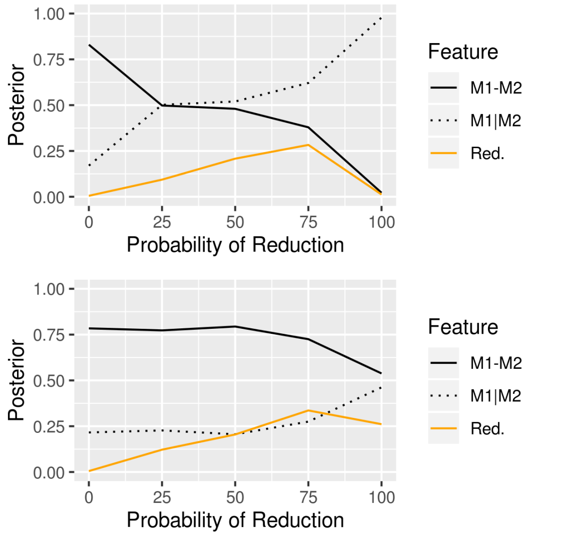


Figure 8: Probability of fusion and vowel reduction in (top) $\mathcal{E}$, (bottom) $\mathcal{F}$ as a function of the probability of vowel reduction.

of vowel reduction. The system always underestimates the true probability of reduction— when the true probability is 50%, for instance, the posterior is only 20%— and counterintuitively, learns that reduction is *absent* when its true probability is 100%. This reflects the influence of the prior bias against the reduction rule, but also the fact that the system learns some cases of reduction as variant lexical items. Table 5 shows one Markov chain's final learned representations for two values of M_1 (*-ta*) and M_2 (*-de*) as a function of reduction rate. With no reduction, the system learns only agglutinative analyses; intermediate systems learn underlying forms for both fused and unfused morphemes, including multiple variant forms of each one. The system with 100% reduction learns only a fused morpheme, *-tte*, which incorporates the result of both vowel reduction and assimilation. With no evidence for an overt vowel between the *ts*, the system has no reason to learn the rule.

Figure 8 (bottom) shows the results for lan-

Rate	M_1=I	M_2=III	M_1=I|M_2=III
0	ta	de	-
25	ta (t, te)	de (te)	tade
50	ta (t)	te (de)	tade
75	ta (ti, t)	te (de)	tte
100	-	-	tte

Table 5: Underlying forms learned for two morphemes in variants of language $\mathcal{E}$. First entry is the posterior mode, (parentheses) show alternatives with $p > .01$.

guage $\mathcal{F}$. As predicted, the probability of fusion increases again with the rate of reduction, but the results are less extreme, since stress placement on the suffixes varies depending on the stem. For this language, agglutination is always the plurality outcome, but intense reduction increases the probability that some fusional analyses will be produced.

Returning to Zingler's argument, Turkish is similar to the case in which the probability of reduction is 0, a case which in our simulations is indeed strongly agglutinative. Because Turkish is syllable-timed and has vowel harmony, it is unlikely to develop the alternate pattern of stress-timing and durational reduction which Zingler argues could lead it to develop more fusion. We have shown that stress-timing and durational reduction does favor fusional analyses. It is tempting to speculate that the same argument might help to explain the differences between Finnish (vowel harmony and agglutination) and Estonian (no harmony and limited fusion); Estonian historically had a more agglutinative structure. In particular, we note that Estonian has word-initial stress (Lippus et al., 2014), which simulation $\mathcal{F}$ shows is predictive of a mixed rather than entirely fusional system.

6 Conclusion

Our results show that, at least in principle, pre-existing typological features can help to determine whether an agglutinative construction evolves into a fusional one, or remains stable. In particular, we present firm evidence that variably-realized marking makes fusion less likely while durational vowel reduction has the opposite effect. While authors like Plank (1999) have listed many independent features or elements which characterize prototypically "fusional" morphology, these have typically been discussed as typological clusters, without necessarily providing a causal explanation. Our modeling results give a mechanism in which some of these features precede, and give rise to, others.

A variety of researchers have noted (Greenberg, 1966) and attempted to discover (Murawaki, 2018; Bjerva et al., 2019) correlations between typological features. Harris (2008) suggests that in many cases, such correlations reflect precisely this kind of historical mechanism— the likelihood that a language will develop in some typological direction is dependent on the features it already has, some of which may encourage a particular change while others tend to reinforce existing patterns. While the simulations presented here use artificial data, we hope to apply this model to real corpus data from languages in which fusion might be developing, in order to isolate particular changes in the phonology as the "triggers" of ongoing morphological change, or explain distributionally why one set of morphemes appears more fusional than another. In doing so, we can discover how theoretical explanations of language change, such as the morphological cycle, might be realized in the minds of language users.

Acknowledgments

We thank the members of the Fall 2018 Seminar on Models of Morphological Learning and Change at OSU for their comments on preliminary versions of this project, Brian Joseph for discussion of historical examples, as well as three anonymous reviewers.

References

R Harald Baayen, Petar Milin, Dušica Filipović Djurdjević, Peter Hendrix, and Marco Marelli. 2011. An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. *Psychological review*, 118(3):438–481.

Johannes Bjerva, Yova Kementchedjhieva, Ryan Cotterell, and Isabelle Augenstein. 2019. A probabilistic generative model of linguistic typology. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1529–1540. Association for Computational Linguistics.

Anthony Brohan and Jeff Mielke. 2018. Frequent segmental alternations in P-base 3. In Larry Hyman and Frans Plank, editors, *Phonological typology*, pages 196–228. de Gruyter.

Joan Bybee. 1997. Semantic aspects of morphological typology. In Joan Bybee, John Haiman, and Sandra A. Thompson, editors, *Essays on language* *function and language type: Dedicated to T. Givón*, pages 25–37. John Benjamins.

Joan Bybee. 1999. Usage-based phonology. In Michael Darnell, Edith A. Moravcsik, Michael Noonan, Frederick J. Newmeyer, and Kathleen Wheatley, editors, *Functionalism and formalism in linguistics, vol. 1: General papers*, pages 211–242. John Benjamins.

Joan Bybee. 2002. *Frequency of use and the organization of language*. Oxford University Press.

Gabriela Caballero and Vsevolod Kapatsinski. to appear. How agglutinative? searching for cues to meaning in Choguita Rarámuri (Tarahumara) using discriminative learning. In Andrea D. Sims, Adam Ussishkin, Jeff Parker, and Samantha Wray, editors, *Morphological typology and linguistic cognition*. Cambridge University Press.

David Chiang, Jonathan Graehl, Kevin Knight, Adam Pauls, and Sujith Ravi. 2010. Bayesian inference for finite-state transducers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 447–455. Association for Computational Linguistics.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago Press.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.

Jan Baudouin de Courtenay. 1876. Rez'ja i rez'jane. *Slavjanskij sbornik*, 3:223–371.

Jonathan Graehl. 1997. Carmel finite-state toolkit. *ISI/USC*.

Joseph H. Greenberg. 1966. *Language universals: With special reference to feature hierarchies*. Walter de Gruyter.

Riho Grünthal. 2007. Morphological change and the influence of language contacts in Estonian. In Hans Fix, editor, *Beiträge zur Morphologie. Germanisch, baltisch, ostseefinnisch, Odense*, pages 403–432. John Benjamins.

Alice C. Harris. 2008. On the explanation of typologically unusual structures. In Jeff Good, editor, *Linguistic universals and language change*, pages 54–76. Oxford University Press.

Martin Haspelmath. 2009. An empirical test of the agglutination hypothesis. In Sergio Scalise, Elisabetta Magni, and Antonietta Bisetto, editors, *Universals of language today*, pages 13–29. Springer.

Martin Haspelmath. 2018. Revisiting the anasynthetic spiral. In Bernd Heine and Heiko Narrog, editors, *Grammaticalization from a typological perspective*, pages 97–115. Oxford University Press.

Bruce Hayes. 1995. *Metrical stress theory: Principles and case studies*. University of Chicago Press.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39(3):379–440.

Jeffrey Heath. 1998. Hermit crabs: Formal renewal of morphology by phonologically mediated affix substitution. *Language*, 74(4):728–759.

Hans Henrich Hock and Brian D. Joseph. 1996. *Language history, language change and language relationship: An introduction to historical and comparative linguistics*. Mouton de Gruyter.

Iván Igartua. 2015. From cumulative to separative exponence in inflection: Reversing the morphological cycle. *Language*, 91(3):676–722.

Martha Booker Johnson. 2015. The beginnings of a sketch grammar on Kihehe. Technical report, Ohio State University.

René Kager. 1995. The metrical theory of word stress. In John A. Goldsmith, editor, *The handbook of phonological theory*, pages 367–402. Blackwell.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70.

Shuan Karim. 2019. Competition between formatives and the diversity of ezafat. In *Proceedings of the 24th International Conference on Historical Linguistics (ICHL24)*.

Valentin Kiparsky. 1979. *Russian historical grammar*. Ardis.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International, Dallas, TX, USA.

Pärtel Lippus, Eva Liina Asu, and Mari-Liis Kalvik. 2014. An acoustic study of Estonian word stress. *Proc. Speech Prosody 2014*, pages 232–235.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Yugo Murawaki. 2018. Analyzing correlated evolution of multiple features using latent representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4371–4382. Association for Computational Linguistics.

David Odden and Mary Odden. 1999. Kihehe syllable structure. In Harry van der Hulst and Nancy A. Ritter, editors, *The syllable: views and facts*, pages 417–446. Walter de Gruyter.

Frans Plank. 1998. The co-variation of phonology with morphology and syntax: A hopeful history. *Linguistic Typology*, 2:195–230.

Frans Plank. 1999. Split morphology: How agglutination and flexion mix. *Linguistic Typology*, 3:279–340.

Jason Alan Riggle. 2004. *Generation, recognition, and learning in finite state Optimality Theory*. Ph.D. thesis, University of California, Los Angeles.

Gregory T. Stump. 1997. Template morphology and inflectional morphology. In Geert Booij and Jaap van Marle, editors, *Yearbook of morphology 1996*, pages 217–241. Kluwer.

Gregory T. Stump. 2001. *Inflectional morphology: A theory of paradigm structure*. Cambridge University Press.

Tim Zingler. 2018. Reduction without fusion: Grammaticalization and wordhood in Turkish. *Folia Linguistica*, 52(2):415–447.

The Role of Linguistic Features in Domain Adaptation: TAG Parsing of Questions

Aarohi Srivastava[1] **Robert Frank**[2] **Sarah Widder**[3] **David Chartash**[4]
Departments of [1]Computer Science and [2]Linguistics,
[3]Program in Cognitive Science, and [4]Center for Medical Informatics
Yale University
aarohi.srivastava|bob.frank|sarah.widder|david.chartash@yale.edu

Abstract

The analysis of sentences outside the domain of the training data poses a challenge for contemporary syntactic parsing. The Penn Treebank corpus, commonly used for training constituency parsers, systematically undersamples certain syntactic structures. We examine parsing performance in Tree Adjoining Grammar (TAG) on one such structure: questions. To avoid hand-annotating a new training set including out-of-domain sentences, an expensive process, an alternate method requiring considerably less annotation effort is explored. Our method is based on three key ideas: First, pursuing the intuition that "supertagging is almost parsing" (Bangalore and Joshi, 1999), the parsing process is decomposed into two distinct stages, supertagging and stapling. Second, following Rimell and Clark (2008), the supertagger is trained with an extended dataset including questions, and the resultant supertags are used with an unmodified parser. Third, to maximize improvements gained from additional training of the supertagger, the parser is provided with linguistically-significant features that reflect commonalities across supertags. This novel combination of ideas leads to an improvement in question parsing accuracy of 13% LAS. This points to the conclusion that adaptation of a parser to a new domain can be achieved with limited data through the careful integration of linguistic knowledge.

1 Introduction

The performance of contemporary syntactic parsers for natural language depends crucially on the availability of training data that matches the sentences on which the parser will be tested. In the realm of constituency parsing, by far the most common corpus used for training is the Penn Treebank (PTB) (Marcus et al., 1993), specifically the subset drawn from the Wall Street Journal (WSJ). It is a truism that the sentences in the WSJ are not an accurate representation of the entirety of English, and indeed the distribution of sentence types in the WSJ differs dramatically from language found in other domains. In particular, interrogative sentences (questions) are quite rare in the WSJ. It is unsurprising, then, that parsers trained on the PTB WSJ corpus perform poorly on questions, sometimes suffering reductions in accuracy of up to 20% (Petrov et al., 2010). However, questions are common elsewhere and indeed are a highly relevant sentence type for a range of NLP applications, such as question answering.

One way to resolve this difficulty involves the dedication of considerable resources to augmenting the training data set with additional hand-annotated parses of the questions. The work reported in this paper explores an alternative method that requires less annotation effort and makes use of three key ideas. First, we follow Bangalore and Joshi (1999) in decomposing the parsing process into two stages: supertagging, where lexically-associated pieces of structure are assigned to each word, and stapling, where these supertags are composed to form a parse tree. Second, we build on the work of Rimell and Clark (2008), where improvements to a supertagger trained with an extended dataset that is less costly to produce lead to improvements in parsing performance using an unmodified parser. However, we find that the parsing benefit that results from improved supertagging can only be maximized when the parser is structured so as to be sensitive to linguistically relevant properties of the supertags. As a result, a necessary third key idea is to use a parser whose input is characterized in linguistic terms that cross-cut the supertag set. This fosters the ability of the parser to generalize across linguistically related, but superficially distinct, sentence types. With the goal of increasing efficiency, following these ideas, a significant increase in parsing accuracy can be seen with a relatively small set of questions for training.

Because we are interested in extracting details of the sentence's interpretation, such as those conveyed through long-distance dependencies, we make use of the Tree Adjoining Grammar (TAG) formalism. TAG is a mildly context-sensitive lexicalized grammar formalism, where the units associated with each word, called *elementary trees*, are pieces of phrase structure that encode detailed information about the word's combinatory potential. Past work (Kasai et al., 2018) has shown that the rich structural representations underlying TAG parsing allow better recovery of long-distance dependencies than is possible with other approaches. Our domain adaptation depends on the rich structure of TAG elementary trees, as we use linguistically-defined features to encode commonalities across trees that the parser can exploit.[1] TAG elementary trees are composed using two operations, substitution and adjoining. The resulting derivations have a structure similar to those familiar from dependency parsing, and indeed computational methods from dependency parsing can be used to accomplish broad coverage TAG parsing (Kasai et al., 2017). As a result, the proposal made in this paper should be more broadly applicable, outside the problem of TAG parsing.

In the first portion of this paper, we introduce the foundations of TAG and the shift-reduce TAG parser employed (Kasai et al., 2017). We then present our methodology of improving the process of assigning elementary trees (supertags) to the words in a sentence to be parsed, and show how and under what conditions improved supertagging can yield substantial benefits for parsing accuracy.

2 Tree Adjoining Grammar

Tree Adjoining Grammar (TAG) (Joshi et al., 1975), is a lexicalized grammar formalism that generates hierarchical structure through a system of tree rewriting. In a TAG derivation, each word in a sentence is associated with an *elementary tree*, a piece of syntactic structure that encodes the structural constraints that the word imposes on the sentence in which it appears. A TAG elementary tree thereby encodes information about the dependencies headed by a word, as well as the structural positions of the word's dependents. For example,

[1]In this respect, TAG is similar to Combinatory Categorial Grammar (CCG) (Steedman, 2000), though the lexical units of CCG carry somewhat less information about structural context, as we will discuss below.

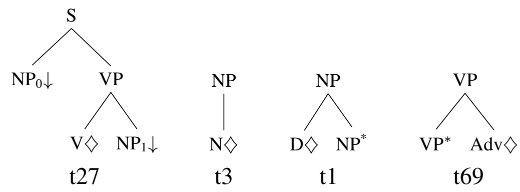


Figure 1: Elementary trees for *Alice read the book quickly*.

a transitive verb like *read* might be associated with the elementary tree t27 on the left of Figure 1, while a name like *Alice* or a noun like *book* would be associated with the elementary tree t3. In these elementary trees, the nodes labeled with the diamond indicate the structural position of the head of the tree. For the verbally-headed tree, the NP nodes that appear along the tree's frontier are the positions for the verb's arguments, i.e., its syntactic dependents. The subscripts on these arguments encode their syntactic relations with the elementary tree's head (0 is subject, 1 is direct object, 2 is indirect object).[2]

Elementary trees are combined using one of two derivational operations: *substitution* and *adjoining*. In substitution, an elementary tree rooted in some category C is inserted into a frontier node in another elementary tree that is also of category C and notated with a down arrow. Thus, to combine the subject NP with the verb in the sentence *Alice read a book*, the NP-rooted elementary tree t3 from Figure 1, headed by *Alice*, is substituted into the NP_0 substitution node in the S-rooted tree t27, headed by *read*.

The second operation, adjoining, introduces recursive structure via a special kind of elementary tree, called an *auxiliary tree*. Auxiliary trees have a distinguished frontier node, the *foot node*, that is of the same category as the root of the tree. The third tree t1 in Figure 1 is an NP-recursive auxiliary tree that would be associated with the determiner *the*. The asterisk on the NP frontier node indicates that it is the tree's foot node. Adjoining works by targeting a node N of category C in some elementary tree using a C-recursive auxiliary tree

[2]These numeric superscripts correspond to "deep" syntactic relations: the subject of a passivized transitive verb will be annotated 1, and operations like dative shift preserve syntactic relations. Though this does not uniquely identify thematic roles of arguments (e.g., unaccusative and unergative subjects are not distinguished), it does provide a richer encoding of predicate-argument dependencies than is provided by usual surface-oriented parses. Recent work has shown that the identity of supertags provides particularly useful information for the task of semantic role labeling (Kasai et al., 2019).

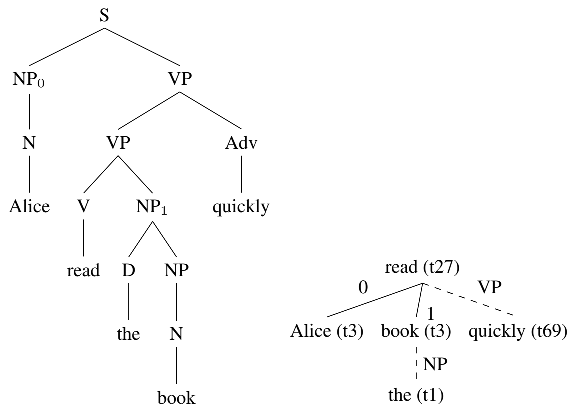


Figure 2: Derived and derivation trees for *Alice read the book quickly*.

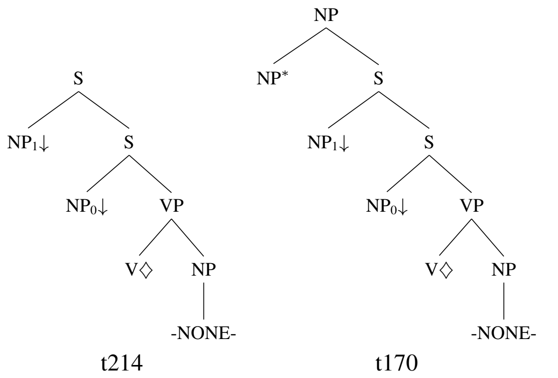


Figure 3: Elementary trees for object questions and object relative clauses.

T. When adjoining applies, the node N is rewritten as the tree T, and N's children are attached (or lowered) as the children of the foot node of T. The determiner tree on the right of Figure 1 can thus adjoin to the NP root of the N-headed tree in the middle of the same figure. In this way, the grammar can generate a structure corresponding to the NP *the book*, which can then be substituted into the NP object substitution node (NP_1) in the transitive verb-headed tree (t27) to derive the entire sentence *Alice read the book*. Similarly, the rightmost elementary tree in the figure, t69, can be adjoined to the VP node in t27 to yield a structure involving adverbial modification. The resulting derived tree structure is given on the left of Figure 2. This derived tree does not, however, represent the derivational steps that were involved in the creation of the structure, which are instead represented in a *derivation tree*. The nodes of the derivation tree correspond to elementary trees, and its edges (dependencies) correspond to substitution and adjoining operations that have applied, i.e., a daughter node is an elementary tree that has been substituted or adjoined into the parent node. Substitution is indicated by solid edges annotated with the index of the substitution site, while adjoining is indicated with dotted edges annotated with the locus of adjoining. The derivation tree for the simple sentence under consideration is given on the right in Figure 2.

TAG shares with the Combinatory Categorial Grammar (CCG) formalism the property of lexicalization: in both formalisms, words are associated with units of structure, elementary trees for TAG and lexical categories for CCG. The presence of rich structure associated with the lexical items is a source of information relevant for a variety of NLP tasks, including semantic analysis and translation, and the use of these formalisms have contributed to performance benefits (Cowan et al., 2006; Xu et al., 2017; Artzi et al., 2015; Nadejde et al., 2017). TAG and CCG differ, however, in the kind of information that the lexical structures encode. In TAG, a verb's elementary tree encodes not only its selected arguments, but also the positions in which they are syntactically realized. Sentences involving long-distance dependencies, such as relative clauses or questions, will therefore involve distinct verbally-headed elementary trees from those used for simple declarative sentences, in which the wh-movement dependency is realized (Frank, 2004). For example, in the question *What did Alice read?*, the displacement of the NP object to the front of the question and its original position filled with a trace node indicated by NONE, as in the Penn Treebank, is represented in the elementary tree t214 on the left in Figure 3. Since the auxiliary verb *did* must appear directly after the fronted NP (NP_1, or *what*, in this case), it adjoins to the S child of NP_1, as shown in Figure 2. In contrast, the verb *read* in the relative clause of the noun phrase *the book that Alice read* would head a different, but related elementary tree, shown on the right in Figure 3, which also includes the fronting of the object, but is itself an auxiliary tree that can adjoin to the NP it modifies.

In contrast, CCG lexical categories do not encode the different realizations of a verb's arguments found in declaratives, questions or relatives. In all such cases, a transitive verb would be assigned the lexical category $(\mathsf{s}\backslash\mathsf{np})/\mathsf{np}$. What differs are the categories assigned to the object (np in simple sentences, $\mathsf{s}/(\mathsf{s}/\mathsf{np})$ for the question word,

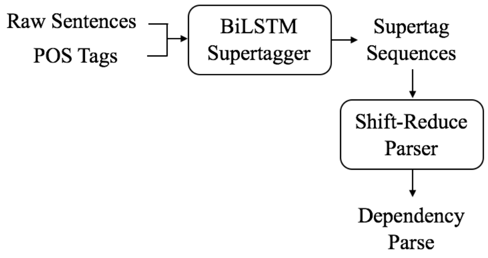


Figure 4: The TAG Parser pipeline.

and (np\np)/(s/np)) for the relative pronoun), as well as the way in which these elements combine with the verb.

3 Supertagging and Parsing

This study uses the TAG supertagger and parser developed by Kasai et al. (2017). The supertagger-parser pipeline is shown in Figure 4. Raw sentences and part of speech tags are given as input to the TAG supertagger, which outputs predicted supertags (i.e., elementary trees) for each word. These predicted elementary trees are given as input to the (unlexicalized) TAG parser, which outputs predicted parses with labeled dependencies among the elementary trees. We briefly review the architecture developed by Kasai et al. (2017). For more details, the reader should consult the original paper.

3.1 Supertagger Architecture

As discussed above, a simple transitive verbal predicate such as *read* might have a different elementary tree depending on the context: t27 as the main predicate of a declarative sentence, or t214 in an interrogative sentence. The same word might have other elementary trees in other constructions, such as subject and object relatives, meaning that the determination of the correct tree requires sensitivity to information that is not local in the string (Kasai et al., 2017). To address the need for long-distance dependency information, the supertagging model makes use of Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997), a recurrent network architecture which is constructed to avoid the vanishing/exploding gradient problem. Specifically, the supertagger developed by Kasai et al. (2017) employs a one-layer bidirectional LSTM network. This architecture processes the input sentence both from beginning to end and from end to beginning. The output of these LSTM units at each time step are concatenated, fed into an affine transformation, and then fed into a softmax unit, yielding a probability distribution over the 4,727 elementary trees that exist in the TAG-parsed corpus we employ, which was extracted from the PTB corpus (Chen et al., 2005). Each word is given to the network as in Kasai et al. (2018): the concatenation of a 100-dimensional GloVe embedding (Pennington et al., 2014), a 5-dimensional embedding of a predicted part of speech tag, and a 30-dimensional character-level representation of the word. The network is trained by optimizing the negative log-likelihood of the observed sequences of supertags.

3.2 Shift-Reduce Parsing Algorithm

Parsing is done using the arc-eager system of shift reduce parsing introduced in the MALT parser (Nivre et al., 2006). This system maintains a stack, buffer, and the set of dependency relations derived so far as the current state. These dependency relations consist of the substitutions and adjoinings that have already occurred between elementary trees. Initially, the buffer holds the sequence of tokens in the sentence, and the transitions terminate when the buffer is empty. At each state, the arc-eager system may choose one of four operations: LEFT-ARC, RIGHT-ARC, SHIFT, and REDUCE, defining ways in which the top elements of the stack and buffer may be manipulated. The TAG parser further divides LEFT-ARC and RIGHT-ARC into seven types according to the derivational operation involved, whether substitution or adjoining, and the location at which the operation takes place (Kasai et al., 2017).

The parser implemented by Kasai et al. (2017) uses a two-level feed-forward network that is trained to predict the operation that should be taken, given the top five elements of the stack and buffer. A noteworthy aspect of the parser is that these data structures contain only supertag information, not the identities of the words in the sentence being parsed. Each supertag is given to the network as a one-hot vector, which is then embedded into a more compact representation, together with vectors that indicate any substitution operations that have already been performed on the supertag. These vector representations of the top elements of the stack and buffer are concatenated and fed to the network, which yields a probability distribution over the possible transition actions. The parser is decoded using a beam search.

3.3 Feature Embeddings

Friedman et al. (2017) explore the benefits of a different input representation for the same parser, involving feature-based embeddings of the elementary trees. These feature-based embeddings are vectors that encode linguistically-defined dimensions of information about the elementary trees specified by Chung et al. (2016). These dimensions include structural properties of the elementary tree (category of the root and head and the category and direction of substitution nodes), subcategorization frame, and grammatical properties (passive, particle shift, wh-movement). The rationale for training a parser with feature embeddings is to allow the network to exploit relationships between trees, and to be able to generalize parsing actions across related contexts. This is particularly useful for cases like passivization and wh-movement, in which the argument structure of the root remains the same, but there are changes in syntax which are reflected in the elementary trees. Friedman et al. (2017) compare the parsing models using both one-hot and featural representations of supertags with respect to parsing performance on PTB sentences, but only saw a "slight improvement" (approximately 0.2% improvement in LAS). However, in the case of adapting to new domains, learning this kind of linguistic information may bridge the gap between the original data domain and the new domain, as it will allow sharing of information about parsing actions for related structures. We explore the importance of providing linguistically-rich feature embeddings to the parser to aid in improving parsing accuracy in the new domain of interrogatives despite never training the parser on sentences from the new domain, especially when limited data is used.

4 Methods

4.1 Background

The most direct approach to adapting a parser for new domains would be to generate a new, hand-annotated dataset that included instances of the new sentence type, which could be used to train a supertagger and parser. Such a process would, however, involve a substantial annotation effort for each new domain. We instead build on the approach of domain adaptation taken by Rimell and Clark (2008). The viability of Rimell and Clark's approach rests on the assumption that "supertagging is almost parsing" (Bangalore and Joshi, 1999). If a parser is provided with a correct set of supertags, it should perform better even on sentence types outside the domain on which it was trained. We therefore focus on retraining the TAG supertagger with a hand-annotated set of questions to which TAG elementary trees have been assigned to each word, but for which parses have not been generated. This hand-annotation process is less expensive than the creation of full parses. As we shall see, this procedure results in improvements in both supertagging and parsing accuracy without ever training the parser on an augmented dataset of questions.

4.2 Data

The question set used in this study contains 350 of the questions used by Rimell and Clark (2008). Their dataset was drawn from the training data provided for the TREC 9-12 Competitions.

4.3 Supertagger Training and Evaluation

To train the TAG BiLSTM supertagger, gold standard part of speech (POS) and supertag sequences were first created for the 350 question set. POS tags were assigned to the 350 questions using the Stanford CoreNLP (Manning et al., 2014) web-based POS tagging tool. These tags were then checked and corrected by hand to create gold standard POS tags.

Next, elementary trees were assigned to the sentences by hand. To make sure these hand annotations were compatible with and followed the same conventions as the method of supertag assignment for the PTB data used to train the parser, the PTB annotation guidelines (Bies et al., 1995) and the gold standard supertag data (Chen et al., 2005) were frequently reviewed. Stanford Tregex (Levy and Andrew, 2006) was used to find relevant trees (e.g., declarative forms of the questions, relative clauses with a similar structure) in the WSJ corpus. Through these methods, ambiguities regarding assignment of elementary trees were resolved. Hand annotation was primarily done by one author, and another author verified or corrected the hand annotations.

In essence, the hand annotation process was conducted as follows. Given the question "What did Alexander Graham Bell (AGB) invent?" the supertag sequence for the corresponding declarative was first determined (Figure 5). From this, the supertag sequence for the question would be created. The biggest change is that the tree for the predicate, *invent*, must reflect the wh-movement (Figure 6).

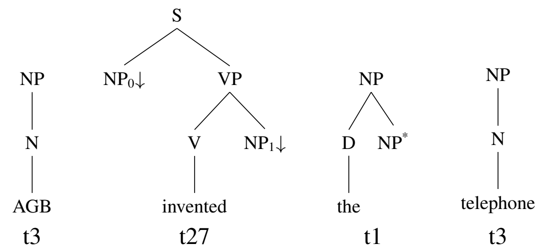


Figure 5: AGB invented the telephone.

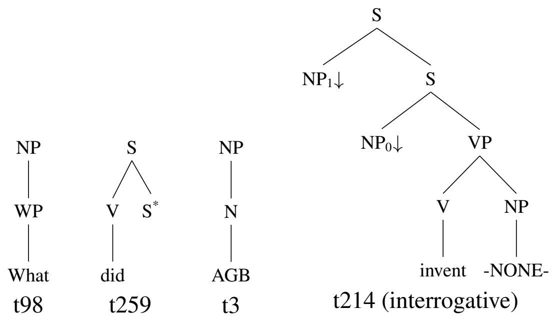


Figure 6: What did AGB invent?

As can be seen, the interrogative elementary tree t214 can be derived from the declarative elementary tree t27. NP_1 has been fronted, and the added auxiliary *did* will adjoin directly after NP_0 at the second S node. Appendix A contains more information about the conventions that were followed in assigning supertags in several common types of questions.

The BiLSTM supertagger was trained with two regimens. In one, only the original PTB training set (WSJ sections 01-22) was provided. In the other, the supertag sequences associated with the hand-tagged questions were added to the PTB data. Rimell and Clark (2008) added ten copies of their 1,328 training questions, adding 13,280 questions to the 39,832 PTB training sentences. Due to the smaller number of hand-tagged questions used for training in this study, 35 exact copies of the training questions were added to the PTB training sentences. This yielded a total of 49,632 sentences in the training set. Through a developmental stage of training and testing, it was determined that 35 copies was optimal to have the highest possible accuracy of supertagging questions without overfitting or reducing accuracy of supertagging PTB sentences. Supertagger training and testing was done using five-fold cross-validation. For each of the five folds, a unique subset of 70 questions was saved for testing, and the remaining 280 questions were used for training. We report mean accuracy over these five folds.

4.4 Parsing Evaluation

In order to analyze parsing performance of questions, gold parses were created for a small test set of 48 questions, each associated with a unique supertag sequence. These questions were not among those used for the training of the supertagger. As before, the assignment of gold parses was done through careful consultation of the PTB annotation guidelines (Bies et al., 1995), as well as the existing TAG-parsed version of the PTB.

For the TAG parser, creation of gold parses requires not only the gold supertag sequences, but also the dependency relations (for UAS and LAS) and the arc labels (for LAS). Two additional columns of information must be added when creating a gold parse as opposed to a gold supertag sequence for a sentence, as shown below. As a result, creating gold supertag sequences is less time-intensive than creating gold parses.

	Word	Supertag	Rel	Arc Label
1	What	t612	2	adjoin
2	continent	t3	5	1 (object)
3	is	t259	5	adjoin
4	India	t3	5	0 (subject)
5	on	t2911	0	root

Two parsing models were explored, both trained only on the PTB TAG parses: (1) the parser model proposed by Kasai et al. (2017) that was trained using one-hot vector embeddings of the elementary trees (henceforth -F), and (2) an identical parser trained with Friedman et al.'s elementary tree feature embeddings (henceforth +F). Decoding for both parsers was done using beam search with a beam size of 16. For each model, three different scenarios were tested, varying in the nature of the supertag input received for the questions to be parsed: (1) supertags given by the original PTB-trained BiLSTM supertagger model (Kasai et al., 2017) (henceforth PTB), (2) supertags given by a supertagger model trained with an augmented dataset of questions and PTB sentences (henceforth PTB+Q), and (3) hand-annotated gold supertags (henceforth Gold). The accuracy of parses in each of the six cases are reported in Section 5.2.

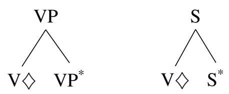
5 Results and Discussion

5.1 Supertagging Results

Supertagging results for the set of 350 questions and the PTB test set are reported separately in Table 1. The PTB-trained supertagger gave an accuracy of 79.61% for the set of 350 questions (an average over the five folds of cross-validation, weighted by the number of words in each fold), and 91.50% for the PTB test set. This PTB-trained supertagger frequently made three types of errors when assigning elementary trees to questions:

1. Incorrect wh-phrase construction: The correct elementary tree for the wh-determiner (e.g., *what* in *what book*) should contain a right NP* adjunction node to adjoin to the NP *book* (as in t1 assigned to *the* in *the book*, Figure 1). Instead, the elementary tree assigned to *book* by the PTB-trained supertagger would incorrectly contain a left NP* adjunction node to facilitate adjunction to the wh-phrase, or the verbal predicate's elementary tree would have two NP substitution nodes into which the wh-determiner and the noun could be inserted separately.

2. Incorrect tree for auxiliary verb: Auxiliary verbs (e.g., *did*) were treated as in a declarative sentence, heading a VP-recursive auxiliary tree t23. Because the auxiliary should appear immediately following the fronted NP and before the subject, the adjunction of the verb should instead take place at S (cf. tree t214 in Figure 3), as in tree t259.



t23 t259

Figure 7: t23 (declarative) vs. t259 (interrogative)

3. Incorrect tree for verbal predicate: The main predicate of the sentence was assigned a declarative elementary tree rather than a question version (i.e., neither fronting nor the NP-NONE trace were expressed in the elementary tree). For a transitive sentence, this means t27 (Figure 1) was assigned to the verbal predicate rather than t214 (Figure 3).

	Questions	PTB Test
PTB training	79.61	91.50
PTB+Q training	95.17	91.64

Table 1: Supertagging Accuracy. Rows indicate training set, whether augmented or not.

		PTB	PTB+Q	Gold
UAS	-F	90.80	90.53	94.60
	+F	91.14	90.51	96.00
LAS	-F	89.63	89.39	94.07
	+F	90.00	89.39	95.81

Table 2: PTB Test Parsing Accuracy. Columns indicate training set for supertagger (or gold supertags) that provide input to the parser. ±F indicates the presence or absence of feature-based supertag embeddings in the input to the parser.

		PTB	PTB+Q	Gold
UAS	-F	81.84	86.70	91.04
	+F	86.18	93.86	99.74
LAS	-F	79.79	85.67	90.53
	+F	83.88	93.09	99.74

Table 3: Question Parsing Accuracy. Columns indicate training set for supertagger (or gold supertags) that provide input to the parser. ±F indicates the presence or absence of feature-based supertag embeddings in the input to the parser.

For the PTB+Q trained supertagger, supertagging accuracy improved, particularly in regards to the three common errors outlined above. On average, supertagging accuracy increased substantially for the question test sets. At the same time, supertagging accuracy on the PTB test set was maintained, indicating that when augmentation is done appropriately, additional training on types of constructions rare in a corpus does not adversely affect supertagging performance on the original corpus.

5.2 Parsing Results

Table 2 reports parsing accuracy on the PTB test set for each of the six parser input conditions described in Section 4 (varying by supertag input and presence or absence of feature-embeddings).[3] We see that the addition of the question data to the supertag's training data (PTB+Q) has a minimal effect on parser performance on the PTB test sentences. Similarly, as found by Friedman et al. (2017), the addition of feature embeddings results in a very small improvement in parsing accuracy, if at all.

[3]Following the standard in the TAG parsing literature, these values do not include accuracy for punctuation.

More relevant for the current topic of discussion is the parsing performance of questions, which is reported in Table 3 for each of the six parser input conditions. We first note that while labeled parsing accuracy (LAS) for the -F parser improved from 79.79% to 85.67% when going from PTB to PTB+Q supertagger training, we see an even more dramatic increase when the feature-trained (+F) parser is used: in this case, parsing accuracy increases to 93.09%. As discussed in Section 3.3, the feature embeddings provide linguistic information over which the parser can generalize from one type of structure to another. Because of the rarity of questions in the PTB, many of the correct supertags used when hand-annotating the question set are also rarely present in the gold standard supertag data for the PTB WSJ corpus (Chen et al., 2005). As a result, the TAG parser (trained only on the PTB WSJ corpus) was not equipped to properly handle these supertags. Thus, while the parsing accuracy increased when given PTB+Q-trained supertags, the improvement is not as large as it might be due to the parser repeatedly encountering uncommon supertags that it was unable to correctly staple together. When the +F parser was used, the parser had learned the knowledge required to better deal with these less common supertags, and parsing accuracy improved from 83.88% to 93.09%. It is notable that this improvement is super-additive: the improvement on LAS (13.3%) is greater than the sum of the individual improvements obtained by using the improved supertagger (PTB+Q) alone (5.88%) or using feature-embeddings (+F) in the parser (4.09%). Thus, we find that with our approach to domain adaptation, when coupled with representations that encode linguistic commonalities across different types of structures, accuracy can increase to a level comparable to the parsing accuracy of the original domain. It is also notable that, when training the supertagger, so few questions (350) are needed to see a significant increase in both supertagging and parsing accuracy (by 15% and 13%, respectively).

Table 4 breaks errors in parsing questions into two categories. The error category of "incorrect wh-phrase" relates to parses of questions that failed to adjoin a wh-determiner to its corresponding noun phrase, or that incorrectly substituted a wh-phrase as an argument of the corresponding predicate. The "missing root" category relates to parses that omit assigning any term in the sentence as the root of the dependency parse, most likely due to complexity or rareness of the correct root word's elementary tree. The number and types of parsing errors deriving from the presence of uncommon supertags in questions (e.g., a parse missing a root) persist in the -F parser. In contrast, these errors are minimal for the +F parser. Treatment of the wh-phrase construction was a specific focus of training the supertagger on questions, and while errors in this category decreased (cf. Table 4) for both parsers once the improved supertags were given, the feature-trained (+F) parser was better able to handle these constructions, and errors decreased much more.

		PTB	PTB+Q	Gold
incorrect wh-phrase	-F	19	9	7
	+F	16	3	3
missing root	-F	16	25	23
	+F	1	0	0

Table 4: Summary of Parsing Evaluation for Questions. ±F indicates the presence or absence of feature-based supertag embeddings in the input to the parser.

It is important to note that, although the number of sentences with a missing root increases from the PTB to PTB+Q trained supertagger, the reason for having a missing root changes. Given the correct (often rarer) supertag for the root in the PTB+Q case, the -F parser is now not equipped to properly combine other trees with it, so the root is skipped. This leads to higher numbers of missing root errors for both PTB and PTB+Q. However, such errors do not occur in the +F parser, as sensitivity to features allows the parser to be better equipped to compose even rare trees correctly. We find then that the statement "supertagging is almost parsing" (Bangalore and Joshi, 1999) is true only when the linguistic content of supertags is known to the parser. When the parser receives correct supertags (gold) and is equipped to handle them properly since it was trained with feature embeddings, it yields near-perfect parses (99.74%).

6 Future Work

We anticipate that the approach of domain adaptation for supertagging and parsing explored here can be applied to other domains. For example, imperatives are another sentence type nearly absent from newspaper corpora, but which are nonetheless a crucial type of input to NLP systems such as

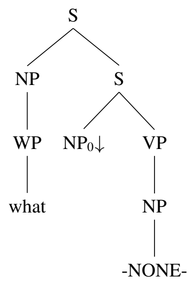


Figure 8: Proposed elementary tree for “what” predicate

virtual assistants. Another domain to which this method might be applied involves biomedical and clinical text (cf. Rimell and Clark 2009), which pose a challenge for information retrieval systems due to the domain-specific vocabulary abbreviations and distinctive syntactic structures, such as null subjects, asyndetic coordination, and fragments.

(a) *abbreviations*: 8 yo M no PMH presents with n/v/F and fever x4 days

(b) *null subjects*: presents with shortness of breath

(c) *asyndetic coordination*: VS notable for fever to 103F, tachycardia, tachypnea

(d) *fragments*: non-toxic though appears ill

In addition, because questions are not well-represented among the original PTB training corpus for the parser, questions on which the parser was tested sometimes involved novel supertags that were absent from the grammar extracted from the PTB. For example, copular sentences with NP predicates (like *Mardi Gras is a festival*) can front the predicate to form a question (as in *What is Mardi Gras?*). The appropriate elementary tree for such cases should be the one given in Figure 8, with the clausal predicate *what* appearing in fronted position. However, no such elementary tree exists among those that were extracted from the PTB by Chen et al. (2005). Consequently, in order to better parse all types of questions, and more generally sentences from other domains, it will be necessary to allow for the creation and feature decomposition of new elementary trees.

7 Conclusion

In this study, we explored an approach to domain adaptation for TAG parsing in the context of questions. We extended Rimell and Clark's approach for improving parsing by improving supertagging. We found first of all that improvements in TAG supertagging, despite the larger number of supertags involved as compared with CCG, are possible through a relatively limited hand-annotation effort. Supertagging accuracy of questions increased by 15%, without sacrificing supertagging accuracy on the original corpus data. Furthermore, while this approach is also successful in improving parsing performance, its effectiveness is maximized when the parser makes use of linguistically-informed representations of supertags. Strikingly, previous work (Friedman et al., 2017) found that the introduction of hand-coded linguistic features in the supertag representations given to the parser does not yield significant benefits in parsing performance. However, our current results suggest that the addition of linguistic features can constitute a crucial source of information when processing structures that are underrepresented in the training data. A parser trained with linguistically-defined feature decompositions of the supertags can better handle those supertags that are uncommon in the data it was trained on. In such cases (e.g., questions), the parser is able to exploit abstract commonalities with related structures, such as relative clauses, that do occur frequently in the training data. Without such linguistically structured representations, considerably more effort would need to be expended to annotate parses in the new domain of questions. We see then that neural methods are not immune to the need for the careful incorporation of hand-coded linguistic features, particularly in addressing problems of domain adaptation.

8 Acknowledgement

The authors are grateful to Jungo Kasai for his crucial advice and technical support throughout this work. We would also like to thank the members of the CLAY lab at Yale, who provided valuable feedback.

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710.

Srinivas Bangalore and Aravind K Joshi. 1999. Supertagging: An approach to almost parsing. *Computational linguistics*, 25(2):237–265.

Ann Bies, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger. 1995. Bracketing guidelines for treebank II style Penn Treebank project. University of Pennsylvania.

John Chen, Srinivas Bangalore, and K. Vijay-Shanker. 2005. Automated extraction of tree-adjoining grammars from treebanks. *Natural Language Engineering*, 12(3):251–299.

Wonchang Chung, Siddhesh Suhas Mhatre, Alexis Nasr, Owen Rambow, and Srinivas Bangalore. 2016. Revisiting supertagging and parsing: How to use supertags in transition-based parsing. In *12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 12)*, pages 85–92.

Brooke Cowan, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 232–241.

Robert Frank. 2004. *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, MA.

Dan Friedman, Jungo Kasai, Thomas R. McCoy, Robert Frank, Forrest Davis, and Owen Rambow. 2017. Linguistically rich vector representations of supertags for TAG parsing. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 122–131. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Aravind K Joshi, Leon S Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of computer and system sciences*, 10(1):136–163.

Jungo Kasai, Robert Frank, R. Thomas McCoy, Owen Rambow, and Alexis Nasr. 2017. TAG parsing with neural networks and vector representations of supertags. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Jungo Kasai, Robert Frank, Pauli Xu, William Merrill, and Owen Rambow. 2018. End-to-end graph-based tag parsing with neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Jungo Kasai, Dan Friedman, Robert Frank, Dragomir Radev, and Owen Rambow. 2019. Syntax-aware neural semantic role labeling with supertags. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: Tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 2231–2234.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Predicting target language CCG supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79, Copenhagen, Denmark. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 705–713.

Laura Rimell and Stephen Clark. 2008. Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 475–484. Association for Computational Linguistics.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, 42(5):852–865.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Pauli Xu, Robert Frank, Jungo Kasai, and Owen Rambow. 2017. TAG parsing evaluation using textual entailments. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 132–141. Association for Computational Linguistics.

A Appendix: Assigning TAG Supertags to Questions

This appendix lays out the linguistic assumptions and analytic decisions that were made for question supertagging and parsing. Within the 350 questions, four basic question types, expressed in a generalized form below, were most common.[4]

a. How many/much ... ?

b. What (NP) is NP ?

c. What (NP) VP ?

d. What (NP) is NP+IN ?

Below we briefly present our assumptions for each type.

How many/much ... ?

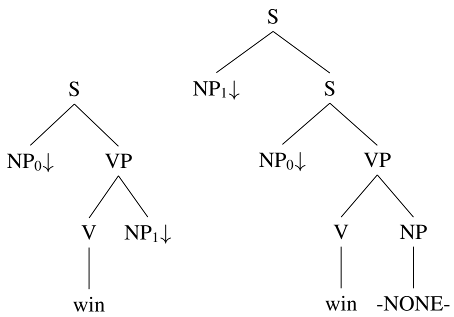
An example of this type of question is:

(1) How many battles did she win?

It is first useful to examine the declarative version closest to this sentence:

(2) She did win five battles.

The key difference between the interrogative version (Sentence 1) and the declarative version (Sentence 2) is the change in order, akin to that of wh-movement. Thus, the elementary tree for the verbal predicate *win* in this question must include the noun phrase trace, as in t214:



t27 (declarative) t214 (interrogative)

As can be seen, the interrogative elementary tree t214 can be derived from the declarative elementary tree t27. NP_1 corresponds to *five battles*. NP_1 in t27 has been replaced by the NP-NONE trace in t214, since it has moved to the beginning of the sentence (fronting). To show this, an additional S node has been added to the top of the tree. Another key difference adopted as a convention is the treatment of *did*. In the declarative sentence, *did* is assigned t23, a VP-recursive auxiliary tree. However, in the interrogative version, *did* is assigned t259, an S-recursive auxiliary tree. The difference is shown below:

t23 t259

This is because of the placement of the additional S node in t214. The auxiliary verb *did* must come between the object (NP_1) and subject (NP_0) of the question, as shown in t214.

What (NP) is NP ?

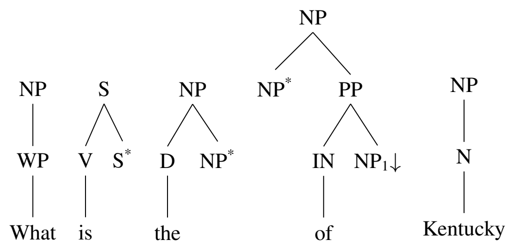
An example of this type of question is:

(3) What is the capital of Kentucky?

with the corresponding declarative sentence:

(4) Frankfort is the capital of Kentucky.

The supertags assigned to Sentence 3 are shown in Figure 9, and the supertag for the predicate is t668 in Figure 10.



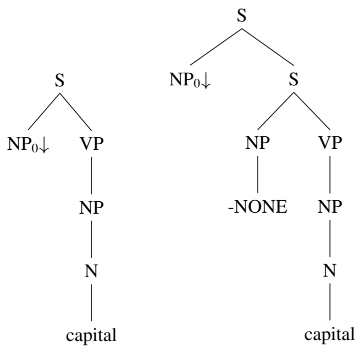
t98 t259 t1 t4 t3

Figure 9: Elementary trees for Sentence 3.

There are two key concepts behind this type of question. First, as for the auxiliary verb *did* in the earlier question type, t23 becomes t259 in the context of questions due to the necessity of adjoining to the S node in a position above the subject. Second, we notice in a copular sentence there is no verb to head the elementary tree, i.e., to project the main S node that serves as the root of the derivation. Instead, the noun *capital* plays the role of predicate of the sentence, and is assigned an S-rooted elementary tree, t668. Figure 10 illustrates the similarity of the two elementary trees assigned

[4]Part of speech tags are taken from the PTB.

to the predicate nominal *capital* in declarative and interrogative forms, with the interrogative t668 encoding the NP-NONE trace.



t167 (declarative) t668 (interrogative)

Figure 10: Elementary trees for "capital" in Sentences 4 and 3, respectively.

What (NP) VP ?

Sentence 5 gives an example of this type of question.

(5) What car company invented the Edsel?

(6) Ford invented the Edsel.

The sequence of elementary trees assigned to this sentence is shown in Figure 11. Although there is no change in word order when converting from the interrogative to the declarative version of this sentence, the verbally-headed elementary tree follows the practice of placing a trace in subject position and displacing the subject to a higher position, as done in the PTB.

Earlier, t214 was used for the question version of the transitive verb *win*'s elementary tree. The difference between t214 and t335 is whether it was the object or subject that was fronted to form the question. Distinct elementary trees are necessary for each possible position of extraction for a given pattern of transitivity.

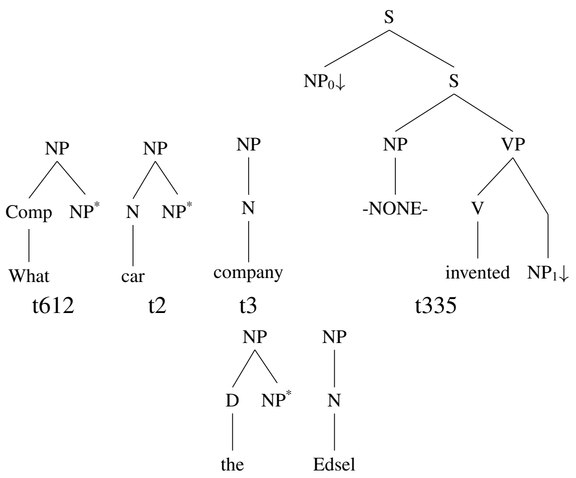


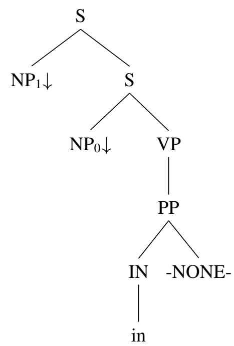
Figure 11: Elementary trees assigned to Sentence 5.

What (NP) is NP+IN ?

The final question type we consider here is as follows:

(7) What city is Logan Airport in?

Unlike copular questions, in which a noun phrase is the main predicate, in Sentence 7 the main predicate is the preposition *in*. As a result, this preposition constitutes the head of the S-rooted elementary tree, as shown in Figure 12, where *what city* substitutes into the NP_1 node (object), and *Logan Airport* substitutes into the NP_0 node (subject).



t2911

Figure 12: Elementary tree used in Sentence 7.

Constraint summation in phonological theory

Benjamin Storme
University of Lausanne
benjastorme@hotmail.com

Giorgio Magri
CNRS
magrigrg@gmail.com

Introduction. *Classical* markedness and faithfulness constraints apply to individual candidates. Yet, the literature has also advocated constraints that instead apply to sets of candidates, such as *Distinctiveness constraints* (DCs; Flemming 2008) and *Optimal Paradigm faithfulness constraints* (OPFCs; McCarthy 2005). These approaches thus need to "lift" the classical constraints to sets of candidates by summing them across a set. Is this assumption of constraint summation typologically innocuous? We formalize this question and establish a positive answer for an *additive* model of constraint interaction.

Constraint summation. DCs embody a preference for more distinct contrasts among SRs: they penalize pairs of SRs whose perceptual distance is below a given threshold. For instance, the DC MinDist in tableau (1) penalizes the pairs of SRs ([ata], [ada]) and ([ada], [a^{n}da]) but not the pair ([ata], [a^{n}da]) because segment pairs ([t], [d]) and ([d], ^{n}d]) are less distinct perceptually than ([t], [^{n}d]), where prenasalization enhances the voicing contrast with the voiceless stop (Flemming 2004).

(1) /ata/, /ada/	MinDist	Id(vce)	Id(nas)	*^{n}D	*D	*VTV
[ata], [ata]	*	*				**
[ata], [ada]	*				*	*
[ata], [a^{n}da]			*	*	*	*
[ada], [ata]	*	**			*	*
[ada], [ada]	*	*			**	
[ada], [a^{n}da]	*	*	*	*	**	
[a^{n}da], [ata]		**	*	*	*	*
[a^{n}da], [ada]	*	*	*	*	**	
[a^{n}da], [a^{n}da]	*	*	**	**	**	

OPFCs embody a preference for greater similarity among SRs in the same morphological paradigm: they penalize pairs of paradigm members that differ along some relevant phonological dimension. For instance, the OPFC IdentOP(length) in tableau (2) penalizes paradigms (b) and (c) because the length of the stem-final vowel is not identical in the two SRs. It does not penalize paradigms (a) and (d), where all the vowels standing in correspondence in the two SRs have the same length (McCarthy 2005).

(2) /faʕaːl-a/, /faʕaːl-tu/	*Vː CCV	IdOP(length)	IdIO(length)
(a) [faʕaːl-a], [faʕaːl-tu]	*		
(b) [faʕaːl-a], [faʕal-tu]		*	*
(c) [faʕal-a], [faʕaːl-tu]	*	*	*
(d) [faʕal-a], [faʕal-tu]			**

DCs and OPFCs are formally very different from classical faithfulness and markedness constraints. In fact, classical constraints assign a number of violations to each individual candidate mapping consisting of a UR and a corresponding SR. DCs and OPFCs instead compare the SRs of multiple candidate mappings. This difference has implications for the architecture of grammar. A "classical" constraint-based grammar evaluates the candidates of a single UR at a time. A grammar with DCs or OPFCs instead must evaluate sets of candidates corresponding to multiple URs, as illustrated in tableau (1) for the two URs /ata/ and /ada/ and in tableau (2) for the two URs /faʕaːl-a/ and /faʕaːl-tu/.

But what about the classical constraints that are now mixed up with DCs and OPFCs? Flemming and McCarthy make the natural suggestion that classical faithfulness and markedness constraints be "lifted" to sets of candidates by *summing* their violations across all candidates in a set. For instance, in (1), the candidate ([ata], [ata]) violates *VTV twice because the two SRs in this pair each violate it once. In (2), the paradigm ([faʕala], [faʕaltu]) violates the input-output faithfulness constraint IdentIO(length) twice because the two SRs in this pair each violate it once.

Typological innocuousness. Tableaux (1)/(2) have two novelties: they contain non-classical constraints such as DCs and OPFCs; and the classical constraints are summed over. Do both novelties contribute to the typological predictions of Flemming's and McCarthy's proposals? In other words, if DCs and OPFCs are left at the bottom, do the classical constraints yield the same winners when they are summed over as when they are used classically for a single UR at the time? Or do the classical constraints make different typological predictions when they are summed as in (1)/(2)?

To formalize this question, we consider two URs (the extension to more than two URs is straightforward). Let A and B be their individual candidate sets, namely the classical tableaux where classical constraints work as usual. Let $<$ be an order over tuples of constraint violations which extends the notion "smaller" from numbers to tuples. We denote by $opt_< A$ and $opt_< B$ the sets of winner candidates in tableaux A and B, namely the sets of those candidates with the "smallest" tuples of violations. We allow $<$ to be a *partial* order, (as needed for HG; see below) whereby $opt_< A$ and $opt_< B$ can contain multiple winners.

Let $A \times B$ be the set of pairs (α, β) of a candidate α in A and a candidate β in B. By Flemming's and McCarthy's constraint summation assumption, a candidate pair (α, β) is represented by the sum $\mathbf{a} + \mathbf{b} = (a_1 + b_1, \ldots, a_n + b_n)$ of the tuples of constraint violations $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ of the two candidates α and β. Tableaux (1)/(2) (without MinDist and IdentOP) illustrate $A \times B$. We denote by $opt_<(A \times B)$ the set of winner pairs in $A \times B$, namely pairs with the smallest summed tuple of violations.

The typological innocuousness of constraint summation relative to a mode of constraint interaction $<$ can be formalized as the identity (3): the two URs considered end up with the same winner candidates if we optimize the product candidate set $A \times B$ relative to the summed constraints (left hand side) or if we optimize the two candidate sets A and B separately (right hand side).

(3) $$\underbrace{opt_< (A \times B)}_{\text{with constraint summation}} = \underbrace{opt_< A \times opt_< B}_{\substack{\text{classical approach}\\ \text{without summation}}}$$

Typological innocuousness in OT. The sum $\mathbf{a}+\mathbf{b}$ carries less information than the two individual tuples of constraint violations $\mathbf{a}$ and $\mathbf{b}$: the individual tuples cannot be reconstructed from their sum. One might thus expect (3) to fail because constraint summation wipes away crucial information. This pessimism is dispelled by an independent result due to Prince (2015): he effectively establishes (3) for the special case where $<$ is OT's lexicographic order. Yet, Prince's reasoning relies on ERCs, a piece of notation tailored to OT. His proof is thus involved because constraint summation does not admit a simple counterpart in ERCs. We show that Prince's result admits the following elementary explanation without ERCs.

Suppose by contradiction that the candidate pair $(\widehat{\alpha}, \widehat{\beta})$ is OT optimal in $A \times B$ but that, say, the candidate $\widehat{\alpha}$ is not OT optimal in A. Hence, there exists another candidate α in A that beats $\widehat{\alpha}$: the tuple $\mathbf{a} = (a_1, \ldots, a_n)$ of constraint violations of α is smaller than the tuple $\widehat{\mathbf{a}} = (\widehat{a}_1, \ldots, \widehat{a}_n)$ of $\widehat{\alpha}$, namely $\mathbf{a} < \widehat{\mathbf{a}}$. Suppose (without loss of generality) that OT's order $<$ is relative to the ranking $C_1 \gg C_2 \gg \ldots \gg C_n$. Thus, $\mathbf{a} < \widehat{\mathbf{a}}$ means (4) holds for some k: the $k-1$ top constraints do not distinguish between the two candidates while the kth constraint decisively assigns less violations to α than to $\widehat{\alpha}$. By adding the corresponding components $\widehat{b}_1, \ldots, \widehat{b}_{k-1}, \widehat{b}_k$ of the tuple $\widehat{\mathbf{b}}$ of constraint violations of candidate $\widehat{\beta}$ to both sides of (4), we obtain (5), which says that $\mathbf{a} + \widehat{\mathbf{b}} < \widehat{\mathbf{a}} + \widehat{\mathbf{b}}$. The candidate pair $(\alpha, \widehat{\beta})$ thus beats the candidate pair $(\widehat{\alpha}, \widehat{\beta})$, contradicting the assumption that the candidate pair $(\widehat{\alpha}, \widehat{\beta})$ is OT optimal in $A \times B$. The proof of the reverse implication is analogous.

(4) $$\begin{aligned} a_1 &= \widehat{a}_1 \\ &\vdots \\ a_{k-1} &= \widehat{a}_{k-1} \\ a_k &< \widehat{a}_k \end{aligned}$$

(5) $$\begin{aligned} a_1 + \widehat{b}_1 &= \widehat{a}_1 + \widehat{b}_1 \\ &\vdots \\ a_{k-1} + \widehat{b}_{k-1} &= \widehat{a}_{k-1} + \widehat{b}_{k-1} \\ a_k + \widehat{b}_k &< \widehat{a}_k + \widehat{b}_k \end{aligned}$$

Typological innocuousness beyond OT. Does the typological innocuousness of the constraint summation assumption extend beyond OT? In other words, besides OT's lexicographic order, which other ways $<$ of ordering tuples of constraint violations satisfy the identity (3)? The crucial property of OT's lexicographic order used in our analysis above is that (4) entails (5): if we add the same quantity to both sides of the inequality, the inequality is not affected. Thus, let us say that an arbitrary order $<$ over tuples of constraint violations is *additive* (Anderson & Feil 1988) provided, whenever a tuple $\mathbf{a}$ is $<$-smaller than a tuple $\widehat{\mathbf{a}}$ and the same tuple $\mathbf{b}$ is added to both, the sum $\mathbf{a} + \mathbf{b}$ is $<$-smaller than the sum $\widehat{\mathbf{a}} + \mathbf{b}$. Hence, (4)/(5) say that OT's lexicographic order is additive. Our main contribution is that **the identity (3) holds if and only if $<$ is an additive order**. In other words, additive orders provide necessary and sufficient structure for the typological innocuousness of the constraint summation assumption. As a corollary, we can extend typological innocuousness of constraint summation from OT to HG.

BLiMP : A Benchmark of Linguistic Minimal Pairs for English

Alex Warstadt,[1] **Alicia Parrish,**[1] **Haokun Liu,**[3] **Anhad Mohananey,**[3]
Wei Peng,[3] **Sheng-Fu Wang,**[1] **and Samuel R. Bowman**[1,2,3]

[1]NYU Linguistics [2]NYU Data Science [3]NYU Computer Science
Correspondence: warstadt@nyu.edu

Introduction & Prior Work We introduce BLiMP (The **B**enchmark of **Li**nguistic **M**inimal **P**airs, or), a large new benchmark dataset for the targeted evaluation of statistical language models' knowledge of linguistic phenomena. The benchmark consists of 67 datasets, each containing 1000 minimal pairs isolating a specific grammatical contrast and collectively offering broad coverage of major phenomena in English grammar. Like the GLUE benchmark for reusable sentence understanding models (Wang et al., 2018), assigns a single numerical score to a language model (LM) measuring its overall mastery of grammar, enabling straightforward comparison of LMs. The dataset is ideal for fine grained analysis of an LM's knowledge of different grammatical domains. For baselines, we evaluate four representative LMs from NLP literature. We find that is hard even for state-of-the-art models, though Transformers perform better than LSTM and n-gram LMs. Humans overwhelmingly agree with the generated minimal pair contrasts in .

A growing body of work evaluates LSTM LMs' knowledge of grammar by testing whether they prefer acceptable sentences over minimally different unacceptable ones (Linzen et al., 2016, a.o.). So far, results have been mixed, motivating the creation of this benchmark which scales up this kind of investigation to isolate dozens of grammatical contrasts within an otherwise-uniform controlled artificial dataset. Our results show that knowledge of grammar has increased as LM technology progressed from n-grams to LSTMs to Transformers. LSTMs and Transformers alike are very accurate in detecting morphological and agreement violations, but state-of-the-art Transformer LMs have an especially large advantage over LSTMs in contrasts where simple generalizations are difficult to find, such as NPI licensing and island effects.

Data consists of 67 datasets of 1000 minimal pairs each, grouped into twelve broader categories (Table 1). A minimal pair consists of two minimally different sentences where one is grammatically acceptable and the other is not. All minimal pairs in contain the same number of tokens and differ only in word order or the identity of one lexical item, following Marvin and Linzen (2018).

We include minimal pairs illustrating linguistic phenomena well known in morphology, syntax, and semantics. While this set is not exhaustive, it does cover a wide range of topics found in formal implementations of English grammar (e.g., HPSG; generative linguistics textbooks). To fully isolate the phenomena of interest, we use realistic artificially-generated sentences, following Marvin and Linzen, a.o. To generate text, we construct a vocabulary of over 3300 lexical items labeled with features reflecting morphology (e.g. singular/plural), syntax (e.g. transitive/intransitive), and semantics (e.g. animate/inanimate), and build a simple artificial grammar for each paradigm.

We validate the acceptability contrasts in the generated pairs with Mechanical Turk annotators, testing 5 randomly-selected pairs from each paradigm using the same forced-choice task models are presented with. Majority vote of 20 annotators agrees with on at least 4/5 examples from each paradigm and on 96.4% of pairs overall.

Baselines We evaluate 4 baselines: (1) An **n-gram** LM trained on the English Gigaword corpus (Graff et al., 2003), based on a modified Kneser Ney implementation by (Heafield, 2011), which considers up to 5-grams, restricting the model from learning dependencies spanning more than 5 words. (2) An **LSTM** recurrent neural network LM from Gulordava et al. (2018). (3) **Transformer-XL** (Dai et al., 2019), a transformer LM with additional features that enable it to model

Phenomenon	N	Acceptable Example	Unacceptable Example
Anaphor agreement	2	*The cats licked themselves.*	*The cats licked itself.*
Argument structure	9	*The cat broke the lamp.*	*The cat vanished the lamp.*
Binding	7	*Bob thinks Ann saw herself.*	*Ann thinks Bob saw herself.*
Control/Raising	5	*The cat is likely to purr.*	*The cat is tough to purr.*
Determiner-Noun agr.	8	*Meg pets those cats.*	*Meg pets that cats.*
Ellipsis	2	*I have a black cat and you have two.*	*I have a cat and you have two black.*
Filler-Gap	7	*The cat noticed the mouse that slept.*	*The cat noticed what the mouse slept.*
Irregular forms	2	*The cat ate the mouse.*	*The cat eaten the mouse.*
Island effects	8	*Whose cat are you petting?*	*Whose are you petting cat?*
NPI licensing	7	*A man who can see Jan hasn't ever left.*	*A man who can't see Jan has ever left.*
Quantifiers	4	*No cat ate more than three treats.*	*No cat ate at least three treats.*
Subject-Verb agr.	6	*The cat that chased the mice sleeps.*	*The cat that chased the mice sleep.*

Table 1: Minimal pairs exemplifying each of the twelve linguistic phenomenon categories covered by . *N* is the number of 1000-example minimal pair paradigms within each category.

model	**Overall**	Ana. Agr	Arg. Str	Binding	Ctrl. Rais.	D-N Agr	Ellipsis	Filler. Gap	Irregular	Island	NPI	Quantifiers	S-V Agr
5-gram	60.5	47.9	71.9	64.4	68.5	70.0	36.9	58.1	79.5	53.7	45.5	53.5	60.3
LSTM	70.8	95.2	73.5	73.2	67.9	84.2	67.3	71.3	92.3	43.9	66.7	62.2	85.1
Transf.-XL	68.7	94.1	69.5	74.7	71.5	83.0	77.2	64.9	78.2	45.8	55.2	69.3	76.0
GPT-2	80.1	99.6	78.3	80.1	80.5	93.3	86.6	79.0	84.1	63.1	78.9	71.3	89.0
Human	88.6	97.5	90.0	87.3	83.9	92.2	85.0	86.9	97.0	84.9	88.1	86.6	90.9

Table 2: Percentage accuracy of four baseline models and raw human performance on using a forced-choice task. A random guessing baseline would give expected accuracy of 50%.

long contiguous inputs of thousands of words during training. (4) **GPT-2** (Radford et al., 2019), a larger neural network LM based on a standard architecture, which is not recurrent and directly models long-distance dependencies.

Our primary evaluation is a forced choice task, in which we test whether a model assigns a higher probability to the acceptable sentence than unacceptable one in each pair. While probability may not correspond to grammaticality when comparing very different sentences, we expect this to be a viable proxy when comparing minimally different sentences as in our data. Additional metrics using word-level probabilities to more narrowly isolate model behavior yield broadly similar conclusions.

Results & Discussion We report model accuracy for the 12 broad categories (Table 2). Overall, the state-of-the-art GPT-2 achieves the highest score and the n-gram the lowest, though all models perform significantly below humans. We find that some phenomena are easier than others: determiner-noun agreement is easy for all models, while islands are quite difficult. We replicate Marvin and Linzen's finding that LSTMs succeed at subject-verb agreement and to some extent binding/anaphora, but largely fail at NPI licensing.

The n-gram model's poor overall performance confirms is not solvable from co-occurrence information alone. Rather, success at is driven by the more abstract (and less interpretable) features learned by neural networks. There are a few exceptions to this pattern: n-grams are mostly sufficient to capture irregular verb forms. Furthermore, SoTA models still show little improvement over n-grams on some phenomena, such as quantifier restrictions and, most strikingly, island effects.

Conclusion We have offered a human-solvable challenge set that covers a broad overview of major grammatical phenomena in English. is hard even for SotA models, though recent large-scale Transformers outperform simple baselines.

References

Z. Dai, Z. Yang, Y. Yang, W. Cohen, J. Carbonell, Q. Le, and R. Salakhutdinov. 2019. TransformerXL.

D. Graff, J. Kong, K. Chen, and K. Maeda. 2003. English gigaword.

K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni. 2018. Colorless green recurrent networks dream hierarchically.

K. Heafield. 2011. KenLM.

T. Linzen, E. Dupoux, and Y. Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies.

R. Marvin and T. Linzen. 2018. Targeted syntactic evaluation of language models.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. Language models are unsupervised multitask learners.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. 2018. GLUE.

Equiprobable mappings in weighted constraint grammars

Arto Anttila
Stanford University
anttila@stanford.edu

Scott Borgeson
Stanford University
borgeson@stanford.edu

Giorgio Magri
CNRS
magrigrg@gmail.com

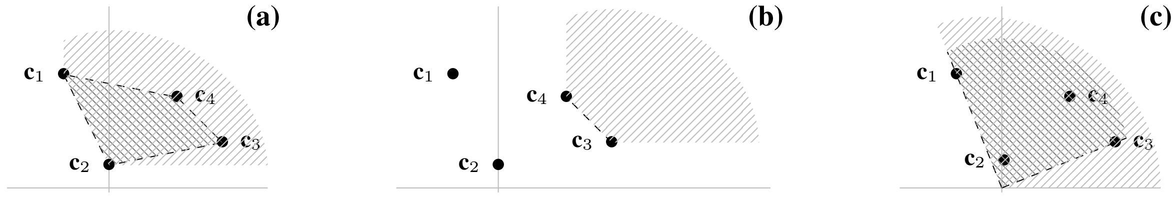
Recent literature (e.g., Smith and Pater 2017) documents cases that admit a better fit in *Maximum Entropy* (ME; Goldwater and Johnson, 2003; Hayes and Wilson, 2008) than in *Stochastic* (or *Noisy*) HG (SHG; Boersma and Pater, 2016). ME is thus richer than SHG. How much richer? This paper addresses this question by comparing ME and SHG in terms of their *equiprobable mappings*.

Equiprobability - A phonological process applies uniformly to all forms that belong to a *natural class* because they share some relevant properties while differing in irrelevant ways. For instance, vowel harmony targets backness but ignores number of syllables. The Finnish mappings (/maa-nä/, [maana]) and (/rakastaja-nä/, [rakastajana]) differ in length, but are *equivalent* for vowel harmony (back). These equivalences are a key property of phonology. How should they be extended to probabilistic phonology? A *probabilistic* grammar assigns to each UR a probability distribution $\mathbb{P}(\mathsf{SR}\,|\,\mathsf{UR})$ over the set of candidate SRs. Two mappings $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are *equiprobable* if every grammar in the typology assigns them the same probability: $\mathbb{P}(\mathsf{SR}\,|\,\mathsf{UR}) = \mathbb{P}(\widehat{\mathsf{SR}}\,|\,\widehat{\mathsf{UR}})$. We submit that equiprobability is the proper way of extending phonological equivalence from categorical to probabilistic phonology. E.g., the fact that words that only differ for length are equivalent for harmony means they have the same probability of harmonizing: $\mathbb{P}(\text{[maana]}\,|\,\text{/maa-nä/}) = \mathbb{P}(\text{[rakastajana]}\,|\,\text{/rakastaja-nä/})$.

ME - Given a winner and a loser mapping, their *difference vector* consists of the constraint violations of the loser discounted by the violations of the winner. Suppose the mapping $(\mathsf{UR}, \mathsf{SR})$ has 4 difference vectors $\mathbf{c}_1, \ldots, \mathbf{c}_4$. The gray region in fig. (a) is their *convex hull*. The lightgray region consists of points larger than a point in this convex hull. Two mappings $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are equiprobable in ME iff they define the same lightgray region. The vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ are *extreme points*: they determine the shape of the lightgray region and must therefore be shared by the two mappings in order for them to share the lightgray regions. The vectors $\mathbf{c}_3$ and $\mathbf{c}_4$ are instead *interior points*: they do not contribute to the shape of the region. Yet, since we have established that $\mathbf{c}_1$ and $\mathbf{c}_2$ are shared, we can effectively "peel them off" the two sides of the ME probability identity. In other words, we can ignore $\mathbf{c}_1$ and $\mathbf{c}_2$ and only focus on $\mathbf{c}_3$ and $\mathbf{c}_4$. They are extreme points of the new lightgray region in fig. (b) and must thus be shared. And so on. In conclusion, the two mappings $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are equiprobable in ME iff they share exactly the same set of difference vectors. Realistically, this happens only if $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are the same mapping. ME thus <u>admits no equiprobable mappings</u>.

SHG - The gray region in fig. (c) is the *convex cone* of the difference vectors $\mathbf{c}_1, \ldots, \mathbf{c}_4$. The lightgray region consists of points larger than a point in this cone. Indeed, the geometry of SHG is analogous to that of ME, with cones in place of convex hulls. Two mappings $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are SHG equiprobable iff they define the same lightgray region. The difference vector $\mathbf{c}_1$ sits on the border but can be shifted (rescaled) without affecting the lightgray region. The equiprobable mapping $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ thus needs *not* share this difference vector $\mathbf{c}_1$ but only a rescaling thereof. Furthermore, nothing can be said in this case about the interior vectors $\mathbf{c}_2, \ldots, \mathbf{c}_4$. In conclusion, the two mappings $(\mathsf{UR}, \mathsf{SR})$ and $(\widehat{\mathsf{UR}}, \widehat{\mathsf{SR}})$ are equiprobable in SHG iff each *non-interior* difference vector is a *rescaling* of a non-interior difference vector of the other mapping. This SHG condition is weaker than than the ME condition above. First, because ME requires *identity* of difference vectors while SHG only requires *rescaling*. Second, because ME looks at *all* difference vectors while SHG ignores *interior* ones. SHG thus <u>does admit equiprobable mappings</u>.

Test case - We test ME's and SHG's predic-



(d)

(kon.sul)(taa.ti.o)ja 0.5%
(kom.mu)(ni.ke.o)ja 0.3%
(o.pe)(raa.ti.o)ja 0.0%
(al.le)(go.ri.o)ja 0.0%

≤

(sym.po)(si.u.me)ja 98.6%
(po.ly)(a.mi.de)ja 95.7%
(in.ku)(naa.be.le)ja 9.5%
(lii.rum)(laa.ru.me)ja 18.6%

≤

(pro.pa)(gan.dis.te)ja 100%
(ak.va)(rel.lis.te)ja 100%

≤

(ter.mos)(taat.te)ja 100%
(mar.ga)(rii.ne)ja 100%
(af.fri)(kaat.to)ja 99.7%

≥

(pro.pa)(gan.dis)(tei.ta) 0.0%
(ak.va)(rel.lis)(tei.ta) 0.0%

≤

(po.ly)(a.mi)(dei.ta) 4.3%
(lii.rum)(laa.ru)(mei.ta) 81.4%
(sym.po)(si.u)(mei.ta) 1.4%
(in.ku)(naa.be)(lei.ta) 90.5%

≤

(al.le)(go.ri)(oi.ta) 100%
(kom.mu)(ni.ke)(oi.ta) 99.7%
(kon.sul)(taa.ti)(oi.ta) 99.5%
(o.pe)(raa.ti)(oi.ta) 100%

(e)

(sym.po)(si.u)(mei.ta) 1.4% ≤ (po.ly)(a.mi)(dei.ta) 4.3% ≤ (lii.rum)(laa.ru)(mei.ta) 81.4% ≤ (in.ku)(naa.be)(lei.ta) 90.5%

(sym.po)(si.u.me)ja 98.6% ≤ (po.ly)(a.mi.de)ja 95.7% ≤ (lii.rum)(laa.ru.me)ja 18.6% ≤ (in.ku)(naa.be.le)ja 9.5%

tions on Finnish secondary stress. In Finnish, (i) primary stress falls on the initial syllable; (ii) secondary stress falls on every other syllable after that (iii) except that a light syllable is skipped if the syllable after that is heavy; (iv) unless that heavy syllable is final (Hanson and Kiparsky 1996). The skipping clause (iii) exhibits probabilistic variation in long words: both (pró.fes.so)(rìl.la) (with skipping) and (pró.fes)(sò.ril)la (without skipping) are attested. The rate of skipping depends on vowel quality and preceding syllable weight (Anttila 2012). Despite secondary stress being hard to hear, Finnish has a segmental alternation that can be used as stress diagnostic: a short underlying /t/ is deleted when *extrametrical*. Thus, skipping correlates with t-retention, as in (pr o.fes.so)(rèi.ta); no-skipping correlates with t-deletion, as in (pró.fes)(sò.re)ja.

To model this distribution of Finnish secondary stress, we constructed an input space consisting of 48 noun types systematically varying stem length, syllable weight, and vowel quality. These phonological forms are evaluated by eight constraints capturing the phonological factors mentioned above. We computed SHG/ME uniform probability inequalities for this model using $\mathbb{C}$o$\mathbb{G}$e$\mathbb{T}$o (available online at `[omitted]`), a suite of $\mathbb{T}$ools for studying SHG and ME based on their rich underlying $\mathbb{C}$onvex $\mathbb{G}$eometry, as illustrated by the results above. SGH predicts seven blocks of equiprobable mappings ordered through uniform probability inequalities (denoted $\leq$) as in fig. (d). This confirms the formal result above that SHG does allow for equiprobable mappings.

To evaluate these predicted equiprobable blocks, we computed the observed t-retention/deletion rates for each stem type in a corpus of approximately 9 million nouns (tokens). The five black SHG-equiprobable blocks are consistent with the data (all stems are nearly categorical), but the two red blocks are problematic. Yet, the difference between t-deletion/retention rates for stems of the liirumlaarumi- and inkunaabeli-type is not statistically significant ($\chi^2 = 2.9849$, $df = 1$, $p = 0.08404$). Furthermore, there are only two stems in the symposiumi-type and both could be re-analyzed as 4-syllable stems, consistently with their high t-deletion rate (Anttila and Shapiro 2017). We have no explanation for the high t-deletion rate for stems of the polyamidi-type ($N = 69$). We conclude that the Finnish data are generally consistent with SHG's predictions.

Does ME offer a more principled treatment of the two problematic red blocks? That is not the case. In fact, as expected given the formal result above, ME breaks up these two red equiprobable blocks and orders their stem types through uniform probability inequalities as in fig. (e). On the retention side (top row), ME seems promising: corpus frequencies mirror the predicted probability inequalities. Yet, on the deletion side (bottom row), ME fails to flip the inequalities, yielding the opposite of what we observe. Such counterintuitive probability reversals seem to recur in ME.

Addendum - OT induces even more equiprobable blocks than HG: it predicts "syllable counting" by grouping together odd-parity stems of different lengths, pointing at a linguistically interesting difference between ranked and weighted constraints.

Script knowledge constrains ellipses in fragments – Evidence from production data and language modeling

Robin Lemke, Lisa Schäfer, Heiner Drenhaus and Ingo Reich
SFB 1102, Saarland University
`robin.lemke@uni-saarland.de`

Abstract

We investigate the effect of script-based (Schank and Abelson 1977) extralinguistic context on the omission of words in fragments. Our data elicited with a production task show that predictable words are more often omitted than unpredictable ones, as predicted by the Uniform Information Density (UID) hypothesis (Levy and Jaeger, 2007). We take into account effects of linguistic and extralinguistic context on predictability and propose a method for estimating the surprisal of words in presence of ellipsis. Our study extends previous evidence for UID in two ways: First, we show that not only local linguistic context, but also extralinguistic context determines the likelihood of omissions. Second, we find UID effects on the omission of content words.

Background In order to communicate a message, speakers can choose between a full sentence (1a) and nonsentential utterances, or *fragments* (Morgan, 1973) (1b). Fragments can convey the same meaning as the corresponding sentence, but lack words that are obligatory in the sentence, like a finite verb. We investigate why people omit particular words in fragments and hypothesize that the choice between omitting and realizing a word is driven by the Uniform Information Density (UID) hypothesis (Levy and Jaeger, 2007), which has been applied to other omissions, like that of relative pronouns (Levy and Jaeger, 2007) and complementizers (Jaeger, 2010).

(1) Ann and Bill are sharing a pizza. She asks:
  a. Would you like another slice of pizza?
  b. Another slice?

Uniform Information Density UID states that *information* is best distributed uniformly across the utterance. Following Shannon (1949), the information, or *surprisal* (Hale, 2001), of a word w_i is defined as the negative logarithm of its likelihood to appear in context (2).

(2) $S(w_i) = -\log_2 p\,(w_i \mid context)$

Surprisal indexes processing effort (Hale, 2001; Levy, 2008), and a uniform distribution makes the most efficient use of the hearer's limited cognitive resources. Previous research has shown that the optional omission of function words reflects optimization with respect to UID (e.g. Levy and Jaeger, 2007; Jaeger, 2010). Optimization consists in two strategies that contribute to a uniform distribution of information: First, omitting uninformative words avoids inefficient local surprisal minima. Second, words that reduce the surprisal of very informative, i.e. unpredictable, following words are more likely to be inserted. If this reasoning also applies to content words like *pizza* in (2), UID can explain why speakers sometimes use a (specific) fragment rather than a sentence: The fragment is preferred over the sentence if it results from omitting predictable words that are obligatory in the corresponding full sentence.

Materials and method Investigating whether omissions are subject to UID requires (i) a set of linguistic data containing the relevant omissions and (ii) surprisal estimates for both the omitted and realized words in this data set. Given these surprisal estimates, logistic regressions can show whether information-theoretic predictors like surprisal affect the likelihood of a word's omission.

Although the term *context* in (2) in principle comprises both linguistic and extralinguistic context (Levy, 2008), most of the previous information-theoretic studies on omissions (like the ones cited above) estimated the surprisal of words from corpora with *n*-gram language models. Such models take only (part of) the linguistic context of the target word into account. However, fragments often occur discourse-initially, so that predictability depends on extralinguistic context that cannot be retrieved from text corpora. Therefore we collected a data set of utterances for tightly controlled script knowledge-based contexts (Schank and Abelson, 1977) with a production task. This data set allows to quantify the effect of both extralinguistic and linguistic context.

Subjects read a story like (3) (original materials in German) and produced the utterance that they considered most likely in that context. Since scripts prime upcoming events (see e.g. Delogu et al., 2018), they should raise expectations about what will be said in a script-based situation. For instance, in (3), a request to pour the pasta into the pot or to give the speaker the pasta is probable.

(3) Annika and Jenny want to cook pasta. Annika has put a pot with water on the stove. Then she has turned the stove on. After a few minutes, the water has started to boil. Now Annika says to Jenny:

In order to use empirically motivated script knowledge representations as stimuli, we based our materials on event chains extracted from DeScript (Wanzare et al., 2016), a crowd-sourced corpus of script knowledge that contains about 100 descriptions of the stereotypical time-course of everyday activities, such as cooking pasta. Following Manshadi et al. (2008), we defined an event as the finite verb and its nominal complement, e.g. `put pot` in (3). After dependency-parsing the corpus (Stanford parser, Klein and Manning (2003)) we extracted these event representations from it. We estimated the likelihood of an event given the previous one with bigram language models trained on the manually preprocessed data for each script with the SRILM toolkit (Stolcke, 2002). We then extracted sequences of three events that were most likely to follow each other and used these event chains to construct our materials. The first sentence in each item introduces the script (cooking pasta), and the next three ones elaborate the event chain (`put pot`, `turn on stove`, `boil water`). For each of 24 items, we collected responses from 100 participants recruited on the crowdsourcing platform Clickworker.

Production data preprocessing As there was a high degree of variation both between scripts and between subjects in the data collected with the production task, we preprocessed the data by manually resolving pronouns and ellipses, lemmatizing the remaining words and finally pooling synonyms to a single lemma. Because we are interested in content words, we removed all function words and adverbials. Removing function words is necessary because e.g. articles and prepositions cannot be freely omitted in standard German (Lemke, 2017; Reich, 2017) and adaptation to UID occurs only "within the bounds defined by grammar" (Jaeger, 2010, 25). Prepositions and distinctive case morphology were annotated on the noun (see (4) for an example), as these features can be important cues towards the meaning intended by the speaker. Adverbials were removed because they can remain implicit in regular sentences and therefore are not involved in the generation of fragments (even though it might be interesting to investigate whether this is subject to UID as well). For the utterance in (4a), preprocessing yields the abstract representation in (4b).

(4) a. Schütte die Nudeln in den Topf!
pour the pasta in the.ACC pot
Pour the pasta into the pot!
b. `pour pasta in.pot`

Investigating the effect of surprisal on omission requires surprisal estimates for both realized and omitted words, therefore we reconstructed all ellipses in the original data. We added those expressions that are minimally required in a full sentence, that is, missing verbs and/or their arguments. This ensures that the outcome of the independent variable, surprisal, is not affected by the dependent variable, omission. The data set for analysis comprises a total of 2.409 sentences consisting in 6.816 primitive expressions ("words" in what follows), 1.052 (15.43%) of these words had been omitted in the original data set.

Surprisal estimation We investigate potential effects of three measures of surprisal: (i) *unigram surprisal*, (ii) *context-dependent surprisal* that takes into account preceding linguistic material within the utterance and (iii) *surprisal reduction*, i.e. how much inserting a word before a target word reduces its surprisal.

We estimate the *unigram surprisal* of each word in the preprocessed data with unigram language models with Good-Turing discount on the preprocessed data that we trained using the SRILM toolkit (Stolcke, 2002). We trained an individual language model on the data for each script separately, because this allows to interpret surprisal as conditioned on the script-based situation, i.e. on the extralinguistic context (5):

(5) $S(w_i) = -\log_2 p(w_i \mid context_{extraling.})$.

We use a novel method based on Hale (2001) to estimate *context-dependent surprisal*, that considers preceding words in addition to extralinguistic context. The default method to quantify effects of linguistic context on surprisal are bigram or higher order *n*-gram models. However, training *n*-gram models on elliptical data brings along a circularity issue observed by Levy and Jaeger (2007, 852): If predictable words are omitted more often than unpredictable ones, their corpus frequency is not proportional to their predictability. This problem could be addressed by ellipsis resolution, but training *n*-gram models on the enriched data set is also not realistic. A trigram model trained on the enriched data set estimates the surprisal of `pot` in a fragment `pour pot`, where `pasta` has been omitted from $p(pot \mid pour\ pasta)$. Crucially, this is psychologically implausible, because `pasta` is not included in the actual linguistic context.

Therefore we estimate context-dependent surprisal (and surprisal reduction, see below) with a method based on the approach by Hale (2001). Hale (2001) derives surprisal from the work done by the human parser, that consists in rejecting all parses that are compatible with the input before but not after processing a word. The larger the total probability mass of the rejected parses is, the more informative is a word. This approach requires to know the likelihood of each parse, i.e. each complete structure, which in our case is equivalent to its relative frequency in the enriched data set. Hale (2001) calculates the surprisal of a word w_i as the log ratio between the prefix probability α, i.e. the total probability mass of the parses compatible with an input, before and after processing w_i:

(6) $S(w_i) = \log \frac{\alpha_{\text{i-1}}}{\alpha_{\text{i}}}$

We modify Hale's approach by allowing for arbitrarily many omissions before and after each word in the input string in order to account for the possibility of ellipses when calculating a word's effect on the set of maintained parses and consequently on α_i. For instance, processing `pour` in the fragment `pour pot` rules out all parses that do not contain `pour`. Processing `pot` now excludes all parses that do not contain `pot` somewhere after `pour`, independently of whether there is a word like `pasta` between `pour` and `pot`. Surprisal is calculated as (6) based on the prefix probabilities before and after these processing steps. Our approach circumvents the circularity issue because it relies on nonelliptical representations. It is also psychologically realistic because it quantifies the work done by the parser incrementally.

Predictors	r^2	t-value	p-value
Unigram, context	.65	70.06	$< .001$
Unigram, reduction	.48	37.99	$< .001$
Context, reduction	.62	54.0	$< .001$

Table 1: Correlations between surprisal predictors.

Finally, we calculate *surprisal reduction*, i.e. how much inserting w_i reduces the surprisal of $w_{i\text{-}1}$, for all non-final words. For this purpose, we calculate the ratio between the prefix probability at w_{i+1} if w_i has been realized and the prefix probability at w_{i+1} if w_i has been omitted. In case of the example, how much the surprisal of `pot` is reduced by inserting `pasta` is calculated as (7).

(7) $S\ reduction(pot{,}pasta) = \frac{\alpha_{put\ ...\ pot}}{\alpha_{put\ ...\ pasta\ ...\ pot}}$

Results We analyzed the data with mixed effects logistic regressions (`lme4`, Bates et al. (2015)) that predict the omission of a word in the enriched data set from the surprisal measures. We first conducted separate analyses of unigram and context-dependent surprisal on the complete data set and then an analysis that considers both unigram surprisal and surprisal reduction for non-final words. In principle it would have been desirable to include all three surprisal measures as predictors in a single regression analysis, but, as table 1 shows, in particular context-dependent surprisal is highly correlated with the other two measures.

The models in the analyses of unigram surprisal[1] and context-dependent surprisal[2] contained by-script random intercepts and slopes for surprisal and by-subject random intercepts. In both analyses there are significant main effects of the respective predictor, that confirm our hypothesis that predictable words are more likely to be omitted. The effect for unigram surprisal ($\chi^2 = 7.39, p < .01$) is stronger than that of context-

[1] Ellipsis ~ UnigramS + (1+UnigramS|Script) + (1|Subj)
[2] Ellipsis ~ ContextS + (1+ContextS|Script) + (1|Subj)

dependent surprisal ($\chi^2 = 4.86, p < .05$).

The analysis that includes surprisal reduction and unigram surprisal[3] was conducted on a subset of the data that contained those non-final words that were not followed by an ellipsis (55.51% of the total data). The final model has random intercepts for subjects and scripts and contains significant main effects of both predictors. The effect of unigram surprisal ($\chi^2 = 10.39, p < .01$) replicates the analysis of the full data set, and the effect of surprisal reduction ($\chi^2 = 27.03, p < .001$) shows that words that reduce the surprisal of the following word more strongly are more likely to be realized. There is no significant interaction between both predictors ($\chi^2 = 0.01, p > .9$).

Discussion Our study confirms the predictions of UID on omissions in fragments: Predictable words are more often omitted in fragments, and words that reduce the surprisal of following ones are more often realized. This extends previous evidence for UID in two ways: First, we find UID effects on the omission of content words. Second, we show that not only local linguistic context, but also extralinguistic context determines the likelihood of omissions. UID however seems not to be the only factor in determining whether fragments are used, as the ratio of fragments varies even between scripts with a similar mean surprisal.

Our study also shows that event probabilities estimated from a corpus of script knowledge provide a reasonable model of extralinguistic context, to which subjects adapt their lingustic behavior. We also propose a method for estimating by-word surprisal in partially elliptical data in a psychologically realistic way. In our study this required a data set that we collected specfically for this purpose and a large amount of manual preprocessing. Future work could show inhowfar our results can be replicated on larger and less constrained data sets when preprocessing steps like reference and ellipsis resolution as well as the standardization of the production data are automatized.

References

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48.

Francesca Delogu, Heiner Drenhaus, and Matthew W. Crocker. 2018. On the predictability of event boundaries in discourse: An ERP investigation. *Memory & Cognition*, 46(2):315–325.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of NAACL (Vol. 2)*, pages 159–166.

T. Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61(1):23–62.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Robin Lemke. 2017. Sentential or not? – An experimental study on the syntax of fragments. In *Proceedings of Linguistic Evidence 2016*. University of Tübingen, online publication system.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Roger P. Levy and T. Florian Jaeger. 2007. Speakers optimize information density through syntactic reduction. In Bernhard Schlökopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press.

Mehdi Manshadi, Reid Swanson, and Andrew S Gordon. 2008. Learning a Probabilistic Model of Event Sequences from Internet Weblog Stories. In *Proceedings of the Twenty-First International FLAIRS Conference*.

Jerry Morgan. 1973. Sentence fragments and the notion 'sentence'. In Braj B. Kachru, Robert Lees, Yakov Malkiel, Angelina Pietrangeli, and Sol Saporta, editors, *Issues in Linguistics. Papers in Honor of Henry and Renée Kahane*, pages 719–751. University of Illionois Press, Urbana.

Ingo Reich. 2017. On the omission of articles and copulae in German newspaper headlines. *Linguistic Variation*, 17(2):186–204.

Roger Schank and Robert Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Enquiry into Human Knowledge Structures.* Erlbaum, Hillsdale.

Claude E. Shannon. 1949. The mathematical theory of communication. In Claude E. Shannon and Warren Weaver, editors, *The Mathematical Theory of Communication*. The University of Illinois Press, Urbana.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado.

Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. DeScript: A crowdsourced corpus for the acquisition of high-quality script knowledge. In *Proceedings of LREC 2016*, pages 3494–3501, Portoroz, Slovenia.

[3] Ellipsis $\sim$ UnigramS * SReduction (1|Script) + (1|Subj)

Information-theoretic characterization of the Sub-regular Hierarchy

Huteng Dai
Rutgers University
huteng.dai@rutgers.edu

Richard Futrell
University of California, Irvine
rfutrell@uci.edu

Our goal is to link two different formal notions of complexity: the complexity classes defined by **Formal Language Theory** (FLT)—in particular, the Sub-regular Hierarchy (Rogers et al., 2013; Lai, 2015; Heinz, 2018)—and **Statistical Complexity Theory** (Feldman and Crutchfield, 1998; Crutchfield and Marzen, 2015). The motivation for exploring this connection is that factors involving memory resources have been hypothesized to explain why phonological processes seem to inhabit the Sub-regular Hierarchy, and Statistical Complexity Theory gives an information-theoretic characterization of memory use. It is currently not known whether statistical complexity and FLT define equivalent complexity classes, or whether statistical complexity cross-cuts the usual FLT hierarchies. Our work begins to bridge the gap between FLT and Information Theory by presenting characterizations of certain Sub-regular languages in terms of statistical complexity.

Statistical complexity theory. Statistical complexity theory deals with stochastic processes: probabilistic models of infinitely-long sequences. For a process X generating sequences of symbols indexed as $\ldots X_{t-2}, X_{t-1}, X_t, X_{t+1}, \ldots$, we define the notation $\overleftarrow{X}$ ("**the past**") to mean $\ldots X_{t-2}, X_{t-1}$, and $\overrightarrow{X}$ ("**the future**") to mean $X_t, X_{t+1}, \ldots$.

The **statistical complexity** of a stochastic process is the minimal amount of information about the past required to faithfully reproduce the future. Suppose that we want to simulate a stochastic process by generating each symbol based on some memory representation M of the past, and that we want to find a memory representation M that simulates the process as well as possible while having minimal information content, measured in bits. This quantity of minimal information is called statistical complexity. Formally, the statistical complexity S of a process X is the minimum entropy of a memory representation M that perfectly simulates the process:

$$S \equiv \min_{M : D_{KL}[\overrightarrow{X}|\overleftarrow{X}||\overrightarrow{X}|M]=0} H[M], \tag{1}$$

where $H[M]$ is the entropy of the random variable M:

$$H[M] \equiv -\sum_x p_M(x) \log p_M(x), \tag{2}$$

and $D_{KL}[\cdot||\cdot]$ is conditional Kullback-Leibler divergence (see Cover and Thomas, 2006), which is zero for identical conditional distributions. Therefore, Eq. 1 indicates the minimum entropy of any memory representation M subject to the constraint that M must allow us to generate a distribution over future sequences $\overrightarrow{X}$ which is identical to the distribution we would have generated given the past $\overleftarrow{X}$.

Further insight comes from considering the different factors that contribute to statistical complexity. Using information-theoretic identities, we break the statistical complexity into two terms:

$$\begin{aligned} S = H[M] &= I[M : \overrightarrow{X}] + H[M|\overrightarrow{X}] \\ &= \underbrace{I[\overleftarrow{X} : \overrightarrow{X}]}_{E} + \underbrace{H[M|\overrightarrow{X}]}_{C}, \end{aligned}$$

where $I[\cdot : \cdot]$ is mutual information, the amount of information in one random variable about another. The term E is called **excess entropy** and quantifies the amount of information in the past which is useful for predicting the future. The term C is called **crypticity** and quantifies the amount of information stored in M which does not end up being useful for predicting the future.

These quantities are easily understood in terms of memory resources used for incremental language production and comprehension. Statistical complexity measures memory load or storage cost; it can be finite even for non-finite-state processes, as long as the sum in Eq. 2 converges. Excess entropy measures integration cost: it says how many bits of information from the past are used when processing the future. Crypticity is the difference between statistical complexity and excess entropy, and measures the amount of information stored in the minimal memory representation M which does not ultimately end up being used to predict the future.

In order to study memory efficiency, we use these quantities to define an efficiency metric, the **E/S ratio**, which is excess entropy divided by statistical complexity. The E/S ratio tells the proportion of bits stored in memory which end up being useful for predicting the future.

Preliminaries. We study Sub-regular languages defined using Probabilistic Deterministic Finite-state Automata (PDFAs). A PDFA is characterized by a set of internal states $\mathcal{Q}$, an alphabet Σ, an **emission distribution** O of symbols $\in \Sigma$ conditional on a state $\in \mathcal{Q}$, a **transition function** $T : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ defining which state the machine transitions into after emitting a symbol, and distinguished initial and final states. In a PDFA, the transition function T is deterministic; in a general Probabilistic Finite-state Automaton (PFA), it may be stochastic, in which case we have a **transition distribution** rather than a transition function. Our indexing convention is: at time t, the PDFA is in state q_t; it generates symbol x_t before transitioning into the next state q_{t+1}. The time indexing convention is shown in Figure 1.

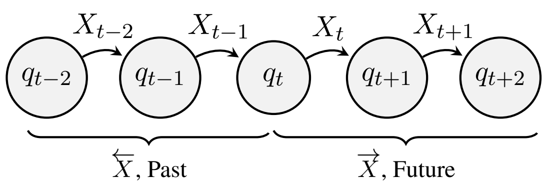


Figure 1: Time-indexing conventions for a finite-state machine.

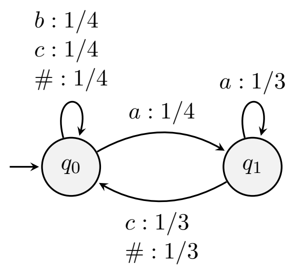


Figure 2: SL_2 PDFA of $\neg ab$, $\Sigma = \{a, b, c\}$

We use the following construction to generate a stationary ergodic stochastic process from a PDFA: whenever the PDFA emits an end-of-word symbol #, it always transitions back into the initial state. The resulting infinite stream of symbols is amenable to analysis using statistical complexity theory. In the literature on statistical complexity, a PDFA of this form is called a **unifilar Hidden Markov Model** (Travers and Crutchfield, 2011, unifilar HMM).

Below, we describe how to calculate S, E, and C from the **minimal trimmed** PDFA (Heinz and Rogers, 2010) for Strictly k-Local (SL_k) languages.

Statistical complexity. For a unifilar HMM, the statistical complexity reduces to the entropy of the stationary distribution over internal states (Travers and Crutchfield, 2011). To get the stationary distribution over internal states Q, we first construct a **state transition matrix**: a stochastic matrix whose entries represent the probability of going into state q_{t+1} after being in state q_t. For a general PFA, the entries of this matrix are given by marginalizing over the emission distribution O:

$$p(q_{t+1}|q_t) = \sum_{x_t \in \Sigma} p_O(x_t|q_t) p_T(q_{t+1}|x_t, q_t),$$

where p_T is the probability of transitioning into state q_{t+1} after generating symbol x_t from state q_t. In a PDFA, this probability is given by the deterministic transition function T, so the transition probability p_T reduces to a Kronecker delta function:

$$p_T(q_{t+1}|x_t, q_t) = \delta_{q_{t+1}=T(x_t,q_t)}.$$

Finally, the stationary distribution over states Q is given by the left eigenvector of the state transition matrix associated with eigenvalue 1.

In general, the statistical complexity of a process depends on the minimal number of states required to represent the process as a PDFA. For an SL_k language, statistical complexity is upper bounded as $S \le (k-1) \log |\Sigma|$.

Excess entropy. For SL_k languages,

$$E = I[X_{t-k+1}, \ldots, X_{t-1} : X_t, \ldots, X_{t+k-2}].$$

In the case of SL_2 languages, we compute E by constructing a **symbol transition matrix**, a stochastic matrix whose entries represent

$p(x_{t+1}|x_t)$, marginalizing over q_t and q_{t+1}. We also need the stationary distribution over symbols, derived from the symbol transition matrix by the same procedure as above.

Crypticity. Crypticity $C = S - E$. In general, crypticity is bounded above by the uncertainty about the emitting state given a symbol:

$$C \leq H[Q_t|X_t],$$

with equality iff X is an SL_2 language.

Sub-regular Hierarchy. We consider two relational structures, namely the successor (+1) and precedence (<) relations. Languages with successor relation keep track of k-long **sub-strings** of the input, such as $\{aa, ab, ac, ba, \ldots\}$ in an SL_2 language. On the other hand, languages with precedence relation keep track of k-long **subsequences**, such as $\{a \ldots a, a \ldots b, \ldots\}$ in an SP_2 language. Different sub-regular languages correspond to distinct PDFAs. For each relational structure, languages with the higher logical power are considered to be more expressive. For example, SL languages are a subset of locally testable (LT) languages. The subset relations are indicated by lines connecting higher and lower regions in Figure 3.

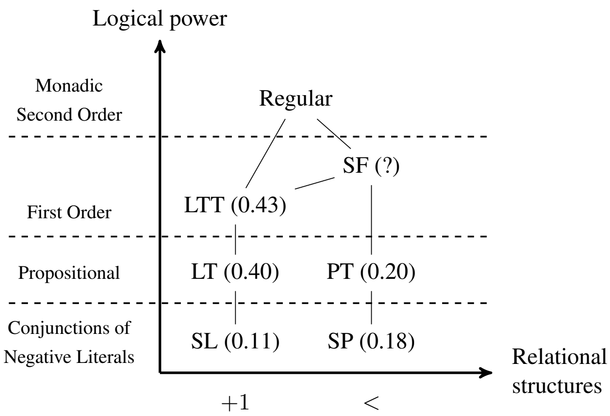


Figure 3: Sub-regular Hierarchy, with E/S ratios calculated from the examples in the text.

Table 1 shows calculated statistical complexity, excess entropy, and crypticity for the minimal trimmed PDFAs of example languages in the Sub-regular Hierarchy, including Strictly Local (SL), Locally Testable (LT), Locally Threshold Testable (LTT), Strictly Piecewise (SP), Piecewise Testable (PT).

The information quantities align with the hypothesis in FLT literature: the languages which

	SL_2	**LT_2**	**LTT_2**	**SP_2**	**PT_2**
Statistical complexity	0.97	1.53	1.94	0.99	1.53
Excess entropy	0.09	$\geq$0.61	$\geq$0.83	$\geq$0.18	$\geq$0.30
Crypticity	0.75	$\leq$0.91	$\leq$1.10	$\leq$0.80	$\leq$1.22
E/S ratio	0.11	$\geq$0.40	$\geq$0.43	$\geq$0.18	$\geq$0.20

Table 1: Information quantities for PDFAs shown in figures. SL_2 = Figure 2; LT_2 = Figure 4; LTT_2 = Figure 5, SP_2 = Figure 6; PT_2 = Figure 7. Quantities marked with $\leq$ or $\geq$ are bounds based on Markov approximations.

are more expressive have higher memory storage requirements. E/S ratios characterize the subset relation in the Sub-regular Hierarchy, for both successor and precedence relations: the higher regions in the hierarchy have higher amount of E/S ratio, as illustrated in Figure 3.

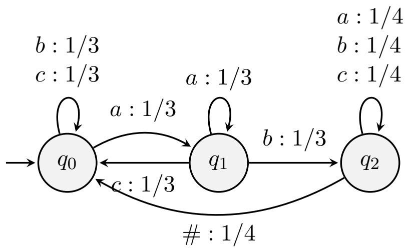


Figure 4: LT_2 PDFA of Some-ab, $\Sigma = \{a, b, c\}$

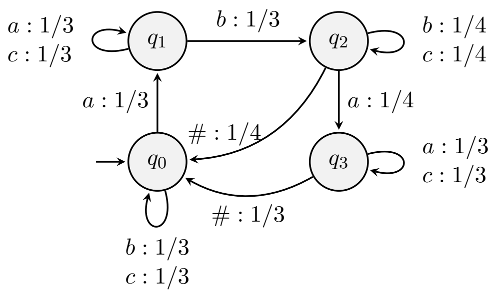


Figure 5: LTT_2 PDFA of One-ab, $\Sigma = \{a, b, c\}$

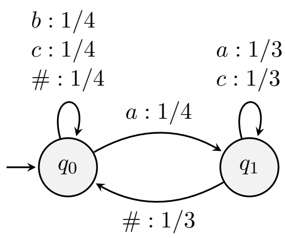


Figure 6: SP_2 PDFA of $\neg a \ldots b$, $\Sigma = \{a, b, c\}$

The information-theoretic characterization illuminates the comparison across relational structures. For example, SL and SP languages correspond to different types of phonotactics: SL

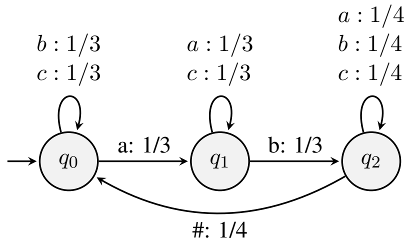


Figure 7: PT_2 PDFA of Some-$a \dots b$, $\Sigma = \{a, b, c\}$

only describes local phonotactics, while SP corresponds to patterns of long-distance agreement. In the examples we have examined, SL and SP have similar information quantities when they share the same k-factor. We conjecture that SL_k and SP_k languages have similar memory efficiency because they are both described by Conjunction of Negative Literals (McNaughton and Papert, 1971, CNL; the combination of $\neg$ and $\wedge$).

Conclusion. We have investigated whether there is a coherent relationship between complexity metrics calculated using Statistical Complexity Theory on one hand, and the Sub-regular hierarchy of languages on the other hand. Our preliminary results, based on example languages representing a number of Sub-regular classes, suggest that increasing logical power corresponds to increasing information-theoretic memory storage requirements. Our current study is limited in that we have only calculated complexity metrics for selected examples of each language class. Future work will work to establish general formal relationships between language classes and statistical complexity.

Regardless of whether statistical complexity turns out to map cleanly onto FLT hierarchies, we believe it provides a promising framework for characterizing bounds on complexity of human languages and phonotactics in particular. The theory of statistical complexity provides a clear way to quantify and reason about memory storage cost and memory integration cost in a highly general information-theoretic setting. Therefore it is entirely reasonable to expect that there may be bounds on the complexity of linguistic subsystems, defined using the language of statistical complexity.

In this connection, we note that statistical complexity depends on a number of factors that are not usually relevant in FLT, such as the transition probabilities and number of states in a PDFA. Although these factors are not relevant in FLT, they may nonetheless be relevant for characterizing constraints on the phonology and phonotactics of human languages. By characterizing complexity using Statistical Complexity Theory, we can take these factors into account in a principled way.

Acknowledgement. We thank Jim Crutchfield, Jeff Heinz, Adam Jardine, and anonymous reviewers for their comments and insights.

References

Thomas M. Cover and J.A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ.

James P Crutchfield and Sarah Marzen. 2015. Signatures of infinity: Nonergodicity and resource scaling in prediction, complexity, and learning. *Physical Review E*, 91(5):050106.

David P. Feldman and James P. Crutchfield. 1998. Measures of statistical complexity: Why? *Physics Letters A*, 238(4-5):244–252.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological Typology, Phonetics and Phonology*, pages 126–195.

Jeffrey Heinz and James Rogers. 2010. Estimating strictly piecewise distributions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 886–896, Uppsala, Sweden. Association for Computational Linguistics.

Regine Lai. 2015. Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry*, 46(3):425–451.

Robert McNaughton and Seymour A Papert. 1971. *Counter-Free Automata (MIT research monograph no. 65)*. The MIT Press.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal grammar*, pages 90–108. Springer.

Nicholas F. Travers and James P. Crutchfield. 2011. Asymptotic synchronization for finite-state sources. *Journal of Statistical Physics*, 145(5):1202–1223.

The Role of Information Theory in Gap-Filler Dependencies

Gregory Kobele
University of Leipzig
gkobele@uni-leipzig.de

Linyang He
Fudan University
lyhe15@fudan.edu.cn

Ming Xiang
University of Chicago
mxiang@uchicago.edu

1 Introduction

Filler-gap dependencies are computationally expensive, motivating formally richer operations than constituency formation. Many studies investigate the nature of online sentence processing when the filler is encountered before the gap. Here the difficulty is where a gap should be posited. Comparatively few studies investigate the reverse situation, where the gap is encountered before the filler. This is presumably due to the fact that this is not a natural class of dependencies in English, as it arises only in cases of remnant movement, or rightward movement, the analysis of which is shakier and more theory laden than the converse. In languages with *wh*-in-situ constructions, like Chinese, the gap-filler construction is systematic, and natural. Sentences (1) and (2) are declarative and matrix/embedded wh-questions respectively in Mandarin Chinese.

(1) LiuBei zhidao CaoCao ai LuBu
LiuBei know CaoCao love LuBu

‘LiuBei knows that CaoCao loves LuBu.’

(2) LiuBei zhidao CaoCao ai shei
LiuBei know CaoCao love who

‘Who does LiuBei know that CaoCao love?’
Or: ‘LiuBei knows who CaoCao loves.’

Although sentences (1) and (2) have similar word order on the surface, in (2) the in-situ wh-phrase *who* takes scope either over the entire sentence (i.e. the matrix question parse) or at the embedded clause (i.e. the embedded question parse). The scope positions precede the wh-phrase, giving rise to the gap-filler dependencies. Gap-filler constructions raise different problems than do filler-gap ones. In the latter, an item is encountered, which needs to satisfy other (to-be-encountered) dependencies to be licensed. There is no uncertainty *that* a gap must be postulated, only *where* it should be postulated. In gap-filler constructions, a dependency is postulated before the item entering into it appears. In contrast to the filler-gap dependency type, gap-filler dependencies do not require more formal power from the syntax; they can (given a finite upper bound on their number) be analyzed with GPSG-style slash-feature percolation and are thus context-free. In systems with (covert) syntactic movement, the *wh*-mover is predictably silent, and could be optimized away (into the context-free backbone of the derivation tree). The motivation for the postulation of a syntactic dependency is to streamline the account of sentence processing; while a purely semantic scope taking account could be implemented (e.g. using continuations), the role and resolution of semantic information during parsing is not as well understood.

Our goal is to understand the role that information theoretic complexity metrics [3] can play in the analysis of Chinese-like *wh*-in-situ constructions. In particular, whether humans' use of probabilistic cues about the presence of a gap can be modeled using the metrics of surprisal and/or entropy reduction. To this end, we identified a sentence processing data set where such cues were manipulated, wrote a Chinese grammar fragment deriving the stimuli, estimated probabilities from the Penn Chinese Treebank 9.0 [8] (using the Stanford NLP Tregrex [4]), and calculated (using the Cornell Conditional Probability Calculator [1]) surprisal and entropy reduction values at each word. Our results show that complexity metrics computed over abstract syntactic structures are significant predictors of processing cost.

2 The data set

We used a data set from an existing eye-tracking reading experiment (Experiment 1 in [7]). The original experiment consisted of 8 different conditions, which were largely designed to create different scoping possibilities for the in-situ wh-word. We implemented the structural properties of these conditions into our grammatical analysis in section 3, such that every condition could be derived by our grammar. An example of half of the original conditions is given in (3a – 3d).

(3a) Matrix Verb Non-predictive; Lower Verb $+Q$

jizhemen zhidao shizhang toulu-le
Reporter know mayor reveal-perf
shizhengfu yancheng-le naxie-guanyuan
city-council punish which-CL-official

"The reporters knew which officials the mayor revealed that the city council punished."
OR "The reporters knew the mayor revealed which officials that the city council punished."

(3b) Matrix Verb Non-predictive; Lower Verb $-Q$

jizhemen zhidao shizhang huangcheng
Reporter know mayor lie
shizhengfu yancheng-le naxie-guanyuan
city-council punish which-CL-official

"The reporters knew which officials the mayor untruthfully claimed that the city council punished."

(3c) Matrix Verb Predictive; Lower Verb $+Q$

jizhemen xiang-zhidao shizhang toulu-le
Reporter wonder mayor reveal-perf
shizhengfu yancheng-le naxie-guanyuan
city-council punish which-CL-official

"The reporters wondered which officials the mayor revealed that the city council punished."

(3d) Matrix Verb Predictive; Lower Verb $-Q$

jizhemen xiang-zhidao shizhang huangcheng
Reporter wonder mayor lie
shizhengfu yancheng-le naxie-guanyuan
city-council punish which-CL-official

"The reporters wondered which officials the mayor untruthfully claimed that the city council punished."

In the 4 conditions above, the wh-in-situ phrase could either take scope at the highest embedded clause or the lower clause. The matrix verb is manipulated. In the *Matrix Verb Predictive* conditions, the matrix verb *wonder* obligatorily take an interrogative complement clause, and therefore in these conditions the wh-phrase is unambiguously high-scope. In the *Matrix Verb non-predictive* conditions, the matrix verb *know* allows an interrogative complement but does not mandate it. The lower embedding verb is also manipulated. The lower +Q verb, such as *reveal*, is in the same class as *know*; but the lower -Q verb, such as *lie*, blocks the lower scope for the wh-phrase since the verb does not allow an interrogative complement clause. The combination of different matrix and embedding verbs yields the scope-ambiguous 3a , and three unambiguous conditions 3b – 3d.

The original experiment contained four additional conditions, all of which were simpler constructions that only contained one embedded clause. The matrix verb was again either predictive or non-predictive of an upcoming interrogative clause. The embedded clause was either short or long with a control verb predicate. An example is given in (4a – 4d).

(4a/b) Matrix Verb Non-predictive or Predictive; Short

jizhemen (xiang-)zhidao shizhang
Reporter know\wonder mayor
yancheng-le naxie-guanyuan
punish-perf which-CL-official

"The reporters knew\wondered which officials the mayor punished."

(4c/d) Matrix Verb Non-predictive or Predictive; Long

jizhemen (xiang-)zhidao shizhang bangzhu
Reporter know\wonder mayor help
shizhengfu yancheng-le naxie-guanyuan
city-council punish which-CL-official

"The reporters knew\wondered which officials the mayor helped the city council to punish."

In this experiment, participants read sentences silently on a computer screen, and their eye-movements were recorded. The data set consisted of data from fifty native Mandarin speakers, each read 48 critical trials based on the 8 experimental conditions.

3 Grammatical analysis

We use the minimalist grammar (MG) formalism [6] to frame our analysis. This formalism allows for the straightforward and transparent encoding of prominent linguistic ideas into a formal system. The lack of support in the CCPC for covert movement pushed us to adopt a feature movement analysis [2] of the Chinese *wh*-in-situ construction, whereby it is not the *wh*-word itself which moves, but rather just a single (*wh*) feature. This is implemented by deriving a *wh*-word by combining a 'pre-wh-word' with a silent (but otherwise overt) *wh*-moving item. This analysis would allow us to implement the observation that *wh*-words in Chinese can be used as well as indefinites, by relating (derivationally) the *wh*-word and the indefinite, although this did not play a role in our analysis.

The analysis encompasses the four clausal complement selecting verb types in the experimental conditions; control verbs ('help'), declarative complement selecting verbs ('believe' or 'lie'), interrogative complement selecting verbs ('wonder'), and verbs which optionally select either declarative or interrogative complements ('know'). Control structures were analyzed in terms of PRO and null case [5], due to CCPC's lack of support for other alternatives. Verbs selecting interrogative complements selected sentential complements, and then immediately checked a *wh* feature. Verbs which select clausal complements irrespective of their force were given two homophonous lexical entries.

The CCPC forces upon us the (computationally motivated) assumption that only one *wh* feature may be active (i.e. moving) at any given time. Thus upon postulating a *wh* 'gap' (i.e. a covert landing site for *wh*-movement), the parser will categorically rule out the (grammatical im-) possibility that a next word is an interrogative complement selector.

4 Frequency Estimation

The CCPC works by translating MGs to equivalent MCFGs, and then parsing using the MCFG. When multiple rules expand the same non-terminal, we need to assign a (non-unit) weight to these rules. As there is currently no MG (or MCFG) TreeBank for Chinese, we were forced to estimate weights of rules by reasoning about the extant structures in the treebank. Due to the small size of our lexicon, there were only five (non-lexical) non-terminals with multiple rules expanding them.

(5a) T[**+WH**]

(5b) VP (with and w/o *wh*)

(5c) AgrO (with and w/o *wh*)

The distinctions relevant to the probability distribution over derivations are not always the ones of obvious interest to linguists. For example, there were two MCFG rules for constructing TPs with *wh*-moving subexpressions. Both rules involve checking the case of a subject DP, but differ as to whether this subject DP is itself **+WH** or **-WH** (in which case the TP necessarily contains another *wh*-word). What we counted in the Treebank is the relative frequency of TPs/Ss which contain active *wh*-words[1] where this *wh*-word is the matrix subject, vs a non-matrix subject. Similarly, a VP can be constructed either by merging an object with a lexical verb, or a derived structure (in this case, necessarily a control verb plus infinitival complement clause). Finally, the category 'AgrO' is the category with which the logical subject is merged (sometimes called 'little-v' in the syntactic literature). The relevant distinctions here (for the non-*wh* case) are whether the AgrO is created by a VP checking the case of its object, or by an interrogative sentential complement taking verb checking the *wh*-feature of its complement, or by a declarative sentential complement taking verb combining with its declarative complement. We counted the relative frequency of transitive verbs (including control verbs) vs interrogative sentential complement taking verbs vs declarative sentential complement taking verbs in the corpus. The relevant distinctions in the case of a **+WH** AgrO are different. A **+WH** AgrO can be created by checking the case of the object of a verb if either the object itself, or some other expression in the VP, is itself **+WH**. Alternatively, it can be created by a *declarative sentential complement* taking verb merging with its sentential complement which contains a +WH expression.

[1] A TP contains an active *wh*-word just in case it contains a *wh*-word which takes scope outside the TP.

The other point of grammatical non-determinism involved the lexicon. Given multiple lexical items with the same featural makeup, we needed to assign weights to the rules which realize a syntactic feature bundle as a particular lexeme. As our lexemes represent whole word classes (*help* stands for the class of control verbs), the only real non-determinism here was in the choice of sentential complement taking verbs (both **+WH** and **-WH**). We counted (for the **-WH** case) the relative frequency with which declarative sentential complements are embedded under *reveal*, *believe*, and *know*,[2] and *mutatis mutandis* for the **+WH** verbs, *reveal*, *know* and *wonder*.

5 Results and discussion

We focused on 4 different eye-movement measures. *First pass duration* is the sum of all fixations in a region from the eyes first entering the region until leaving it either to the left or to the right. *Go-past* time is the sum of all fixations from first entering a region until leaving the region to the right, including fixations made during regression to earlier parts of the sentence. *Second pass duration* is the sum of all fixations in a region following the initial first-pass fixations. *Total time* is the overall reading time (all fixations) in a given region. For each eye-movement measure, we computed average reading time (RT), collapsing over participants and trials, for each word region under each condition. Next using the CCPC software, the grammar analysis in section 3 and the frequency estimation in section 4, we generated the entropy reduction (ER) and surprisal predictions for each word region under each condition. We then performed four linear regressions, using ER and surprisal as predictors and the four eye-movement measures as dependent variables.

Neither ER or surprisal are significant predictors for the first pass duration (*ps*$>$.5). For the go-past time, surprisal is not significant ($p>.2$), but ER is ($p<.05$). However, the model with ER as a predictor accounted for very little of the overall variance in the data (adjusted R^2=0.04). For second-pass and total time RTs, both ER and surprisal are significant (*ps* for ER $<.01$; *ps* for surprisal $<.001$). When both predictors are considered in the same model, R^2=0.23 for the second pass measure and R^2=0.32 for the total time measure. When the two predictors are considered separately, surprisal accounted for more variance in the data than ER (R^2=0.17 surprisal vs. 0.05 ER for the total time; 0.13 vs. 0.03 for the second pass time).

If we consider the four eye-movement measures *first pass*, *go past*, *second pass* and *total time* form a scale to measure effects from the earlier stages of processing to the later stages, we observe that for the current data set information-theoretic complexity metrics such as ER and surprisal seem to mostly explain later measures but not the early ones. With the second pass and total time measures, although ER and surprisal seem to have only accounted for a relatively small amount of variance in the data (with surprisal having a better performance than ER), the current results nonetheless demonstrate the independent effect of abstract structure in parsing, decoupled from effects based on lexical information.

[2]Not the ratio of declaratives vs interrogatives embedded, but, given that a declarative is embedded, how frequently it is embedded under one of these vs the others.

References

[1] Zhong Chen, Tim Hunter, Jiwon Yun, and John Hale. Modeling sentence processing difficulty with a conditional probability calculator. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

[2] Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, MA, 1995.

[3] John T. Hale. Information theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412, 2016.

[4] Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *LREC*, pages 2231–2234. Citeseer, 2006.

[5] Roger Martin. Null case and the distribution of PRO. *Linguistic Inquiry*, 32(1):141–166, 2001.

[6] Edward Stabler. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer, 1996.

[7] Ming Xiang and Suiping Wang. Locality and expectation in mandarin wh-in-situ dependencies. manuscript under review. 2019.

[8] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238, 2005.

Questioning to Resolve Transduction Problems

Eric Meinhardt,[†] Anna Mai,[†] Eric Baković,[†] Adam McCollum[‡]
[†]UC San Diego, [‡]Rutgers University

Elgot & Mezei (1965) show that any non-deterministic regular function (NDRF) $\phi : \Sigma^* \rightarrow \Gamma^*$ can be decomposed into the composition $\rho \circ \lambda$ of two subsequential functions (SSQs) that proceed in opposite directions; crucially, the first function to apply λ must behave as unbounded lookahead for the second. We henceforth refer to such decompositions $\rho \circ \lambda$ as 'EM decompositions'. Recent work in computational phonology has shown the utility of such decompositions for analyzing and comparing the minimum expressivity required for iterative, bidirectional, (non-)myopic, and other long-distance phonological processes that require greater expressivity than that supplied by SSQ functions. Existing work has identified the (interaction-free) weakly deterministic functions (IF-WDRFs; McCollum et al. 2018, Hao & Andersson 2019) and the NDRFs as salient lower and upper bounds on the complexity of such processes (Heinz & Lai 2013, Jardine 2016, McCollum et al.). Because unbounded lookahead is a key feature of this region, we suggest that understanding it is crucial for picking out additional phonologically interesting subclasses within this region. In this work, we identify several concepts useful for describing lookahead in decomposed NDRFs and offer a set of necessary and sufficient properties for a composition $\rho \circ \lambda$ to be an EM decomposition of a non-SSQ NDRF ϕ. We then use these ideas to outline a set of functions in between the IF-WDRFs and proper NDRFs, organized in terms of a precise notion of the degree of lookahead that λ can provide for ρ.

For present purposes,[1] a *question* may be identified with a partition Q over a set of possible worlds W (e.g. a formal language L) into equivalence classes ('cells'), and a *resolving* answer or observation is information that picks out (with respect to some background knowledge — e.g. prior knowledge of L and information gleaned from an observed prefix of a current input string) the cell q_k of the partition that the actual world (total string, unseen suffix, etc.) falls into. While two distinct answers a_i, a_j may resolve a question in the same way by picking out the same cell, entailment defines a (partial) ordering on the *informativeness* of answers or observations: if a_i and a_j pick out the same cell q_k, but a_i is strictly more specific than a_j, then $a_i \vDash a_j$ but both *resolve* Q in the same way. Similarly, *refinement* can be used to define an analogous ordering on questions: if every cell of Q_0 is a subset of some cell of Q_1, then any resolving answer to Q_0 is also a resolving answer to Q_1. An agent faced with choosing the next action sequence ('output string') $u \in \Gamma^*$ given its current knowledge about the state of the world is faced with a *decision problem* that induces a partition on W: each cell is associated with the ('optimal') action sequence that the agent should take at the current timestep if it thinks the actual world currently is in that cell.

A non-SSQ NDRF ϕ at some point while reading the prefix x of a string xy faces a(t least one) 'decision problem':[2] exactly what the incremental output of the prefix x should be depends on which of at least two cells q_k, q_l some *a priori* unboundedly distant portion of the as-yet unseen suffix y falls into. Consider the hypothetical 'sour grapes' pattern entertained by McCollum et al., based on Turkish and dubbed 'Zurkish': [+round] spreads left to right from initial U, changing I to U, unless there is a low vowel A anywhere in the word, in which case there is no spreading at all.[3] Thus input strings of the form UI^n are mapped to UU^n, but input strings of the form $UI^*A^+X^*$ ($X = \{I, A\}$) remain unchanged. Whether a given prefix $x = UI^n$ maps to UU^n or to UI^n depends on whether the suffix y

[1]These concepts are adapted from literature on the *meaning of questions* and the *value of questions and information* (see e.g. van Rooy 2003), but no familiarity with such literature is necessary.
[2]We have not yet considered multiple decision problems per NDRF ϕ, especially incomparable ones.
[3]In actual Turkish, [+round] spreading proceeds up to A, which blocks further spread.

contains an A. If $\rho \circ \lambda$ is an EM decomposition of ϕ, then it must be the case that λ reads input strings xy from the far end relative to ρ,[4] identifies which cell the suffix y belongs to, remembers this long enough to recognize where within x it should transform the input string (be it via markup symbols, length-increasing codes, or 'phonotactic' codes; McCollum et al., Smith & O'Hara 2019), and creates an intermediate string $\lambda(xy) = x'y'$ such that reading the transformed prefix x' from the other end is sufficient to *resolve* ϕ's *decision problem* — i.e. identify which cell the suffix of the original string belongs to and therefore what output string should be emitted. Thus in hypothetical Zurkish, λ reads input strings from right to left and ρ reads the output of λ from left to right. If the suffix y contains an A, then λ transforms the input string such that all instances of I between A and the beginning of the string are marked to not be changed by ρ; otherwise, all instances of I after initial U will in fact be changed by ρ. This thus resolves ϕ's decision problem for Zurkish. A further constraint on λ's rewriting is that ρ must be able to recognize this transformed prefix and thereby infer the associated cell at a particular point in time, viz. by the time it reads the input symbol (or within an *a priori* bounded distance after) associated with ϕ's decision problem. Finally, ρ's output for the symbol associated with the decision problem must then *depend on* the information about y that λ has injected into x'.

Our work synthesizes the results of Elgot & Mezei with those of McCollum et al. and Heinz & Lai. First, we explicate the notions of 'information smuggling' and lookahead left informal in McCollum et al.'s discussion of 'interacting' compositions; thus equipped, we can formally articulate for any non-SSQ $\phi \in$ NDRF the properties that *any* potential EM decomposition $\rho \circ \lambda$ must have in order for it to suffice as an EM decomposition of ϕ. Second, it follows clearly and explicitly from our analysis of EM decompositions that the IF-WDRFs $\subsetneq$ NDRFs. Third, we conjecture that the framework we present offers a useful way of defining and comparing functions with more expressivity than the interaction-free WDRFs but less than the full set of NDRFs. We sketch our current model of such functions below.

In this hierarchy of 'lookahead-constrained' ('LoCo') weakly deterministic regular functions,[5] interaction is possible, but the 'questions' the lookahead pass λ in an EM decomposition can 'answer' for ρ are qualitatively constrained in some way — e.g. λ might be OSL or I-TISL (Hao & Andersson). For any two potential lookahead functions f, g, we can ask whether the question partition of one is a refinement of the other. We conjecture that this can be extended to classes of functions to compare how relatively fine or coarse the questions each can answer when employed as a lookahead function in an EM decomposition. Finally, we can also use the analysis of EM decompositions described above to identify substrings where $\rho \circ \lambda$ interact, but where the change in behavior of ρ on a given substring cannot be be associated with a strict increase in knowledge about the unseen suffix.

References. • Elgot, C. & J. Mezei. 1965. On relations defined by generalized finite automata. *IBM Journal of Research and Development* 9(1). 47–68. • Hao, Y. & S. Andersson. 2019. Unbounded Stress in Subregular Phonology. In *SIGMORPHON 16*, 135–143. ACL. • Heinz, J. & R. Lai. 2013. Vowel harmony and subsequentiality. In *MoL 13*, 52–63. • Jardine, A. 2016. Computationally, tone is different. *Phonology* 33(2). 247–283. • McCollum, A. G., E. Baković, A. Mai & E. Meinhardt. 2018. The expressivity of segmental phonology and the definition of weak determinism. *lingbuzz/004197* . • van Rooy, R. 2003. Questioning to resolve decision problems. *Linguistics and Philosophy* 26(6). 727–763. • Smith, C. & C. O'Hara. 2019. Formal characterizations of true and false sour grapes. In *Proceedings of SCiL 2019*, vol. 2 1, 338–341.

[4] For clarity, we use 'prefix' here from the view of ρ: iff $w = xy$ and ρ sees x first, x is a prefix.

[5] To be precise: we can define a *bounded lattice* (organized by refinement of questions) of non-SSQ NDRFs, with IF-WDRFs at the bottom, otherwise-unrestricted NDRFs at the top, and LoCo WDRFs in between.

Quantifier-free tree transductions

Shiori Ikawa Akane Ohtaka Adam Jardine
Department of Linguistics
Rutgers University
{shiori.ikawa,akane.ohtaka,adam.jardine}@rutgers.edu

1 Introduction

In this work in progress we discuss issues in extending *quantifier-free (QF) logical transductions* from strings to trees. Input-Strictly-Local (ISL) functions, which form an effective class to describe phonological transformations (Chandlee, 2014; Chandlee and Heinz, 2018) and for projecting tiers for long-distance well-formedness conditions (De Santo and Graf, 2019) have been shown to be characterizable with order-preserving QF transductions (Chandlee and Jardine, 2019). We explore how QF transductions can be extended to trees for the purpose of capturing syntactic phenomena. We show QF tree transductions are incomparable to existing tree transducer classes, but do capture some empirically useful transductions. Also, they may be extended with least-fixed point logics to capture a wider range of phenomena, as has been shown for QF logics in strings (Chandlee and Jardine, 2019).

2 Formal definitions

2.1 Logical transductions

Following Courcelle (1994) and Engelfriet and Hoogeboom (2001), we define transductions as logical interpretations. A *signature* is some set of named functions and relations, and a (finite) *model* in that signature is an instantiation of those functions and relations over some (finite) universe of elements. A transduction from models in one signature to models in another can then be described by defining the relations and functions in the output signature using formulas in a logical language of the input signature.

More specifically, for trees labeled with an input alphabet Σ, we define a function to trees over an output alphabet Γ with a series of monadic predicates $\varphi^c_\gamma(x)$—written in the first-order logic of the input trees, without quantifiers—for each $\gamma \in \Gamma$ and $c \in \mathscr{C}$, where $\mathscr{C}$ is a *copy set* that allows us to build $\texttt{card}(\mathscr{C})$ copies for each element in the input tree. The semantics of a transduction is then that an element t in the input tree has a corresponding element labeled t^c in the output tree if and only if $\varphi^c_\gamma(x)$ is true for t.

2.2 Quantifier-free transductions over trees

As a running example for QF tree transductions, we will use the tier-construction function for case assignments. Vu et al. (2019) analyze case assignment as a local well-formedness condition over a tree 'tier', which is itself a tree with irrelevant information removed. The ungrammaticality of the sentence "*He saw she", is captured with a tier constructed by removing all information except D heads carrying NOM or ACC features, C heads, and their immediate parent nodes, as shown in Figure 1: This sentence is bad because the resulting tier contains the local configuration [• he [• she]], where no C head intervenes between the two NOM-featured D-heads as shown in Figure 1.b. Such tier construction functions are non-capturable with simple eraser function (Heinz et al., 2011), as they refer to the input local context in deciding whether to project a certain node. TSL over this tier is more parallel to the Input-local TSL (ITSL) defined over strings in De Santo and Graf (2019), which utilizes the local information in the construction of tiers by constructing tiers with ISL functions, i.e. QF transductions.

There are several considerations required in extending QF logical transductions to trees. First, in order to capture local information with monadic predicates, QF string transductions were defined in Chandlee and Lindell (forthcoming) and Chandlee and Jardine (2019) using functional signatures, where the element in a string are ordered with predecessor and/or successor functions. For our QF tree transductions we assume an input sig-

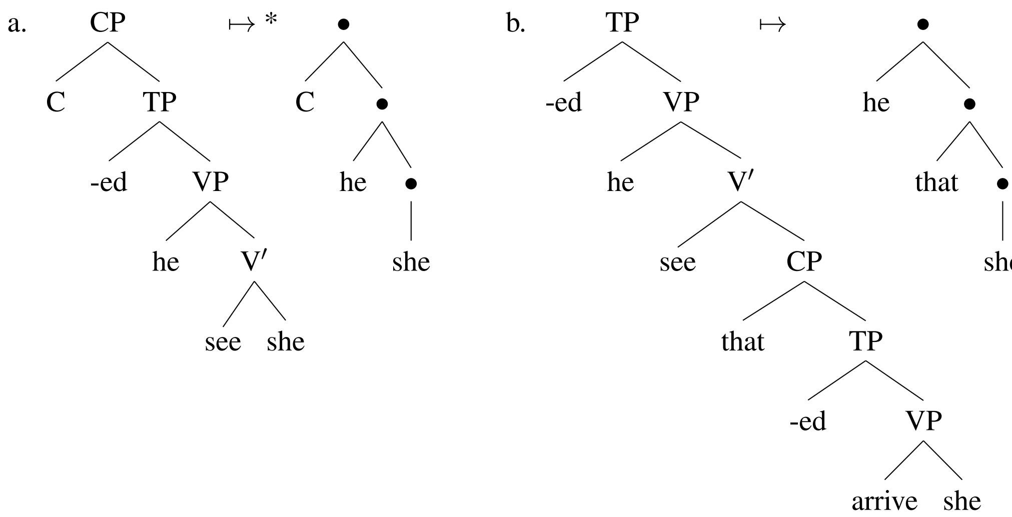


Figure 1: Caption

nature with a parent function μ, where $\mu(x) = y$ when y is the parent node of x, and the predecessor function p, which defines the linear order between sister nodes. Note that we do not use the child relation (i.e. the inverse of μ function), as it is not a function. This means that in (1) we cannot identify the mother nodes of C and D nodes without existentially quantifying the child nodes, so we instead build two copies of C and D nodes themselves.

Second, whereas $\mathscr{C}$ is taken from an initial segment of the natural numbers for string transductions, our copy set $\mathscr{C}$ forms a tree. Members of $\mathscr{C}$ are marked with Gorn address, where the Gorn address of the root will be r. Additionally, exactly one $c \in \mathscr{C}$ will be marked as a 'bottom node' with an additional b label. Every copy tree has to include an r node and b node, as characterized by the well-formedness conditions for a copy tree in (1): When a node exists, the nodes above it including the root node exist (1a) and when a root exists, there is always a bottom node (1b). We will assume that there is at most one root copy r and one bottom copy b. Note that b is a copy to which the lower part of the input tree attaches to, and it does not mean b has to be the lowest node inside $\mathscr{C}$. An example for a copy tree is given in (2a). The case-tier transductions can now be characterized as shown in (2b) and (2c), using the copy tree of the form in (2a).

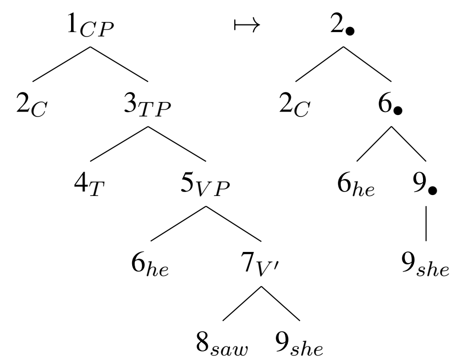
(1) copy well-formedness conditions
For nodes c' and d' s.t. $d' <_\mu c'$,

a. $\varphi_D^{c'}(x) \rightarrow \varphi_D^{d'}(x)$

b. $\varphi_D^{r}(x) \rightarrow \varphi_D^{b}(x)$

(2) a. 

b.
$$
\begin{array}{lcl}
\mathscr{C} & := & \{rb, 0\} \\
\varphi_\bullet^{rb}(x) & := & C(x) \vee he(x) \vee she(x) \\
\varphi_C^{0}(x) & := & C(x) \\
\varphi_{he}^{0}(x) & := & he(x) \\
\varphi_{she}^{0}(x) & := & she(x)
\end{array}
$$

c. 

2.3 Asymmetric c-command preservation

In a parallel way to how order-preservation in string QF transductions restricts them to regular functions (Filiot, 2015; Chandlee and Jardine, 2019), we will define the structural relationship among the output copies in a way that preserves the structural relation of the input tree: We define the output dominance relation based on the asymmetric c-command in the input, as shown in Table 1a (p. 4): As for the input node x, y $s.t.$ (i) y is dominated by x or (ii) y is asymmetrically c-commanded by x ($higher(x, y)$) and x's parent node and sister node that dominates y, are deleted ($sa\text{-}del(x, y)$), the nodes above bottom node of the copy tree of x dominate all the nodes of the copy tree of y. The latter case serves to keep the asymmetric c-command relation between x and y when the intermediate nodes are deleted. In the copy of the same input node, the domination among nodes is trivially defined. Table 1b shows that the precedence relations in the input trees are preserved among the root nodes of the correspondent copy trees in the output, and the precedence relation among the copies of the same input node is defined trivially.

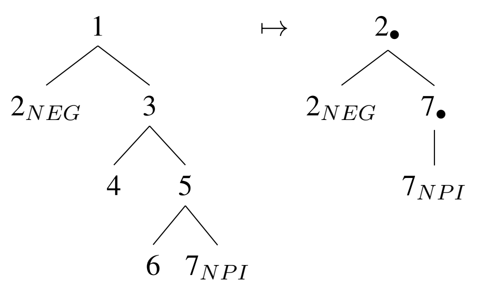
2.4 Comparison with other tree transducers

In general, QF tree transductions as defined here are incomparable to deterministic bottom-up or top-down tree transducers (Comon et al., 2008). Briefly, this is because QF tree transductions get a finite "lookahead" in either direction. However, for this reason, QF tree transductions have some similarities to sensing tree automata (Martens et al., 2008; Graf and De Santo, 2019). Future work will examine this relationship further.

3 Other Examples

3.1 Negative polarity tier construction

The definition of tree transductions discussed above can accommodate the case of negative polarity item (NPI) licensing in English. An NPI such as *anyone* is licensed when it is c-commanded by a downward entailing operator such as negation, as the contrast between "John doesn't like anyone" and "*Anyone doesn't like John" shows. The grammaticality of the sentence "John doesn't like anyone" can be captured with a tier of the form in (3). Crucially, just like the case-tier transduction in (2c), the NPI-tier transduction in (3) is QF-definable using the copy tree in (2a), as shown in (4) (see also Graf and Shafiei 2019).

(3)


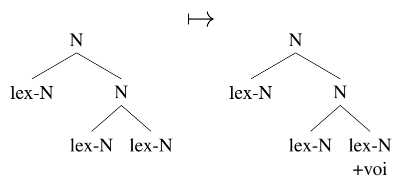
(4)
$$\begin{array}{lcl} \mathscr{C} & := & \{rb, 0\} \\ \varphi^{rb}_{\bullet}(x) & := & NEG(x) \lor NPI(x) \\ \varphi^{0}_{NEG}(x) & := & NEG(x) \\ \varphi^{0}_{NPI}(x) & := & NPI(x) \end{array}$$

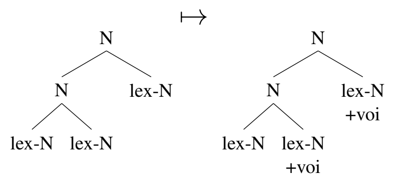
3.2 Morphological conditioning of *rendaku*

Applicability of tree transductions extends to phonological phenomena as well. Japanese has a phonological operation called *rendaku*, where the first consonant of the second element gets voiced in compounding (e.g. *ao* 'blue'+*sora* 'sky' → *ao-zora* 'blue sky'). There is a structural constraint in this operation in (5) (Otsu, 1980): *sora* does not get voicing when it is in the compound [*ao*-[*sora-mame*]] 'blue broad-bean.' Compounds of the structure [[A B] C] allows their second element to undergo *rendaku* (e.g. [[ao-zora]-yohoo]'forecast of blue sky')

(5) Branching Constraint
Rendaku does not occur on B when the compound has the structure [A [B C]].

The application of a [+voi] feature to a structure can be represented as tree transductions in (6). These transductions are QF-definable as shown in (7): The lex-N node which is not *first* (i.e. the left-most among its sisters) acquires [+voi] feature.

(6) a.


b.


(7) a. $\varphi^{rb}_{\mathrm{N}}(x) := \varphi_{\mathrm{N}}(x)$
b. $\varphi^{rb}_{\text{lex-N}}(x) := \varphi_{\text{lex-N}}(x) \land \text{first}(x)$
c. $\varphi^{rb}_{\text{lex-N(+voi)}}(x) :=$
$\varphi_{\text{lex-N}}(x) \land \neg\text{first}(x)$
where $\text{first}(x) := p(x) \approx x$

This pattern cannot be captured by (functional) string transductions: Given a string of three lex-N, we cannot decide between the mappings in (8a) and (8b).

(8) lex-N lex-N lex-N
a. ↦ lex-N lex-N(+voi) lex-N(+voi)
b. ↦ lex-N lex-N lex-N(+voi)

For all $c, d \in T_{\mathscr{C}}$ and b and r of $T_{\mathscr{C}}$,

a.	$x <'^{c,d}_{\mu^*} y :=$	$x <_{\mu^*} y \vee (higher(x,y) \wedge \neg\varphi^b_D(\mu(x)) \wedge sa\text{-}del(x,y)$	$c \leq_{\mu^*} b$
		$x \approx y$	if $c <_{\mu^*} d$
	where	$higher(x,y) := \mu(x) <_{\mu^*} y \wedge \neg\mu(y) <_{\mu^*} x$	
		$sa\text{-}del(x,y) := \neg\exists z[sisters(x,z) \wedge \varphi^b_D(z) \wedge z <_{\mu^*} y])$	
b.	$x <'^{c,d}_{p^*} y :=$	$x <_p y$	if $c = d = r$
		$x \approx y$	if $c <^*_p d$

Table 1: Formulas for preserving asymmetric c-command

Note that it is not always the case that both of these outputs are grammatical given an input string of three nouns. The examples above illustrate: *ao-zora-yohoo* 'foreceast of blue sky' but *ao-sora-mame* 'blue broad-bean' (cf. **ao-zora-mame*).

4 Future work

Chandlee and Jardine (2019) discuss extending QF logic with least-fixed point operators to capture long-distance processes; a clear next step is to extend this to QF tree transductions. Additionally, for n-branching trees we can study their models with a set of n child functions, instead of the mother function used here.

Finally, as already mentioned, the connection between these logical characterizations and sensing tree automata is a likely place to look for direct connctions between logical and automata-theoretic transductions.

References

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee and Jeffrey Heinz. 2018. Strictly locality and phonological maps. *LI*, 49:23–60.

Jane Chandlee and Adam Jardine. 2019. Quantifier-free least fixed point functions for phonology. In *Proceedings of the 16th Meeting on Mathematics of Language*, pages 50–62. ACL.

Jane Chandlee and Steven Lindell. forthcoming. A logical characterization of strictly local functions. In Jeffrey Heinz, editor, *Doing Computational Phonology*. OUP.

Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. 2008. Tree automata techniques and applications. Published online: http://www.grappa.univ-lille3.fr/tata. Release from November 18, 2008.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Proceedings of Formal Grammar 2019*, pages 35–50.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

Emmanual Filiot. 2015. Logic-automata connections for transformations. In *Logic and Its Applications (ICLA)*, pages 30–57. Springer.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on Mathematics of Language*, pages 12–26. ACL.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of SCiL 2019*, pages 205–215.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Wim Martens, Frank Neven, and Thomas Schwentick. 2008. Deterministic top-down tree automata: Past, present, and future. In *Proceedings of Logic and Automata 2008*, pages 505–530.

Yukio Otsu. 1980. Some aspects of rendakuin Japanese and related problems. In Ann Farmer and Yukio Otsu, editors, *MIT Working Papers in Linguistics*, pages 207–228. Cambridge, MA: Department of Philosophy and Linguistics, MIT.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.

Graph-to-Graph Meaning Representation Transformations for Human-Robot Dialogue

Mitchell Abrams[1], Claire Bonial[2], and Lucia Donatelli[3]

[1]Georgetown University, Washington DC
[2]U.S. Army Research Laboratory, Adelphi, MD
[3]Saarland University, Saarbrücken, Germany
mja284@georgetown.edu

Introduction. This research forms part of a larger project focused on natural language understanding (NLU) in the development of a two-way human-robot dialogue system in the search and navigation domain. We leverage Abstract Meaning Representation (AMR) to capture and structure the semantic content of natural language instructions in a machine-readable, directed, a-cyclic graph (Banarescu et al., 2013). Two key challenges exist for NLU in this task: (i) how to effectively map AMR to a constrained robot action specification within a particular domain; and (ii) how to preserve necessary elements for general understanding of human language with the goal that our robot may expand its capabilities beyond a single domain. To address these challenges, we establish a two-step NLU approach in which automatically-obtained AMR graphs of the input language are converted into in-domain meaning representation graphs augmented with tense, aspect, and speech act information. Here, we detail both rule-based and classifier-based methods to transform AMR graphs into our in-domain graphs, thereby bridging the gap from unconstrained natural language input to a fixed set of robot actions.

Background: Data & Annotations. To determine the type of language found in our task and how it is represented in AMR, we used a corpus of human-robot dialogue in which a person directs a remotely located robot to complete search and navigation tasks (Marge et al., 2016). We then manually selected 504 utterances made up of short, sequential excerpts of the corpus data that are representative of the variety of common exchange types that we see. These sentences were independently double-annotated (IAA 87.8% using the Smatch metric (Cai and Knight, 2013)) and adjudicated following current AMR guidelines.[1]

Notably absent from current AMR representation and essential to our task are two elements: (i) tense and aspect information; and (ii) speech act information regarding speaker intent. To address (i), we adapted the annotation system of Donatelli et al. 2018 for tense and aspect; see Bonial et al. 2019 for details. The absence of speech acts in AMR was anticipated, as existing AMR corpora are text-based.[2] For our task, however, an off-the-shelf taxonomy of speech acts was not ideal. The language found in our domain generally adheres to the division of *information-transfer* and *action-discussion* found in other dialogue act classification systems for conversational agents (e.g., Bunt et al. 2012), yet it also tends to group into specific categories related to our robot's abilities and the search-and-navigation task.

We therefore developed a set of 27 template-like AMRs specific to the task of human-robot dialogue, inspired by classical work on speech acts (Austin, 1975; Searle, 1969). These augmented AMR templates are skeletal AMRs in which the top, anchor node is a fixed relation corresponding to a speech act type (e.g., `assert-02` in the AMR lexicon); one of its numbered arguments, or 'ARGs', is a fixed relation corresponding to an action (e.g., `turn-01`) or the content of the speech act; and arguments of these relations are filled out to detail both dialogue relationships (utterance level) and action specifications (content level) (Bonial et al., 2019). These 27 speech acts are classified into 5 types, listed in Fig. 1 (number of subtypes in parentheses), along with example subtypes for the type `command`. Tense and aspect information are currently annotated only on the content level.

As an example of how our augmented AMRs work, a template for `command:move` is shown in Fig. 2(b); in Fig. 2(c), this template is filled in with the specifics of the utterance *Move to the wall.* Fig. 2(a) shows the original AMR. Note, although

[1] https://github.com/amrisi/amr-guidelines/blob/master/amr.md

[2] https://amr.isi.edu/download.html

SPEECH ACT TYPES

```
c / command (6) → | command:move
a / assert (9)    | command:turn
r / request (4)   | command:send-image
q / question (3)  | command:repeat
e / express (5)   | command:cancel
                  | command:stop
```

Figure 1: Speech act types with example subtypes.

```
(a) (m / move-01 :mode imperative
        :ARG0 (y / you)
        :ARG1 y
        :ARG2 (w / wall))
(b) (c / command-02
        :ARG0-commander
        :ARG1-impelled agent
        :ARG2 (g / go-02 :completable +
            :ARG0-goer
            :ARG1-extent
            :ARG3-start point
            :ARG4-end point
            :path
            :direction
            :time (a / after
                :op1 (n / now))))
(c) (c / command-02
        :ARG0 (c2 / commander)
        :ARG1 (r / robot)
        :ARG2 (g / go-02 :completable +
          :ARG0 r
          :ARG3 (h / here)
          :ARG4 (w / wall)
          :time (a2 / after
                :op1 (n / now))))
```

Figure 2: The utterance *Move to the wall* represented in (a) AMR form, (b) domain specific bare template form, and (c) as a filled-in domain specific graph.

absent in the utterance itself, our template captures key information such as start point and who is addressing whom. It also generalizes across utterances related to movement: whether the instruction uses the word *move, drive* or *proceed*, the in-domain representation is the same. The original AMR captures any lexical differences. The template-like structure further helps identify any critical missing information that may prohibit the robot from successfully completing a given action with required roles and aspectual annotation that specify the existence of an achievable goal (`:completable` ±; see Bonial et al. 2019 for discussion).

To establish a gold standard set of in-domain graphs, two authors manually transformed and adjudicated a subset of 290 single-sentence utterances from the larger human-robot dialogue corpus of 504 AMRs described earlier.

Graph-to-Graph Transformations. We convert AMRs, such as that seen in Fig. 2(a)[3], into our in-domain graphs (e.g., Fig. 2(c)) through a mixed methods approach of both rule-based and classifier-based systems, outlined in Figure 3. Following this transformation pipeline, the system requires both the original AMR and original natural language utterance as input. From the utterance, classifiers first determine the speech act and tense information. The classified speech act then triggers one of the corresponding templates. The speech act subtype is identified by matching the root action predicate in the original AMR to any predicates in a dictionary of keywords associated with each subtype. Aspectual information is triggered by specific patterns of speech act and tense combinations. Next, regex searches the original AMR to extract additional relevant arguments and action predicates that correspond to slots in each template, transforming them when necessary (e.g., *you* to *robot*). Details on each step follow.

While there exists a neural AMR graph converter for a related task (Liu et al., 2015), neural systems require substantial training data in the form of annotated input and output graphs. In contrast, our partially rule-based approach leverages the highly structured AMR information and a relatively small data set of natural language text with speech act or tense labels to train the classifiers. Additionally, our two-step approach, in which we maintain both the original parsed AMR as well as the augmented in-domain AMR, allows us to keep track of both the sentence meaning determined by the linguistic signal alone, and the speaker meaning particular to our context (Bender et al., 2015).

Speech Acts. A speech act classifier predicts one of the five speech act types from the original utterance, triggering the appropriate in-domain template for use. Since natural language is variable, we implement a classifier that will be robust to any language input, rather than rely on a rule-based approach in this step. We implement an off-the-shelf Naive Bayes multinomial classifier for our baseline from the scikit-learn library, using unigrams as features (Pedregosa et al., 2011).[4]

In order to classify the speech act subtype (e.g., `command:move`, `command:turn`), the pipeline uses regex to find the root predicate

[3]We plan to obtain AMRs using automatic parsers including Lindemann et al. 2019.

[4]Though we explored using *unigrams*, *bigrams*, and *unigrams + bigrams*, unigrams performed best as our domain is fairly restricted and predictable from individual words. Higher-order n-grams were not effective due to sparsity issues from a small training set and introduced noise into our system.

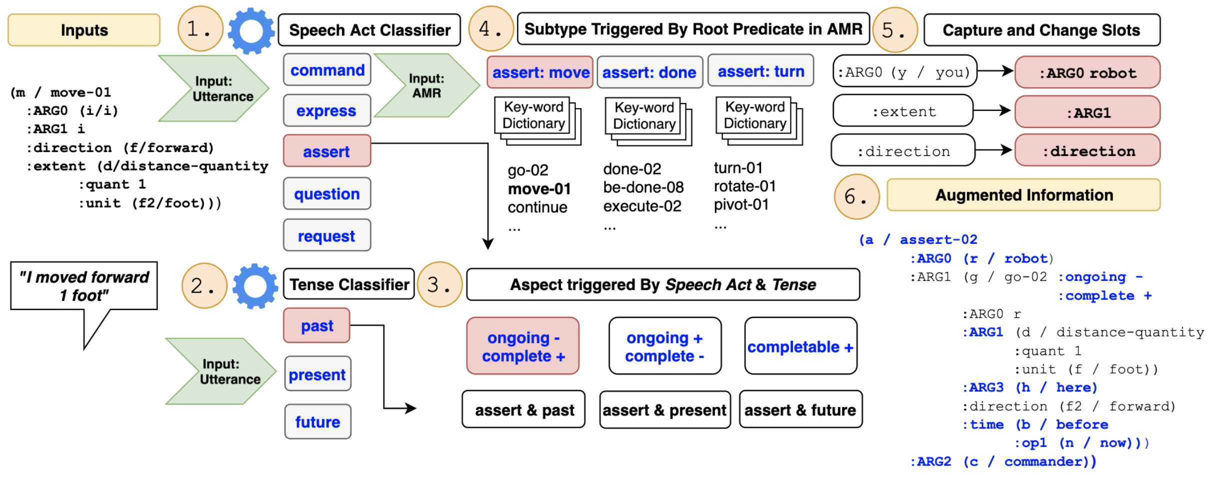


Figure 3: Mixed methods approach to graph transformation. Classifiers and rule-based systems collect and store relevant slot information from the original AMR and original utterance to output an augmented AMR

in the original AMR. In our domain, only a small group of predicates correspond to specific subtypes within each speech act category. For instance, assuming an utterance is classified as a `command`, if `go-02` or `move-01` is found to be the root predicate of the original AMR graph, then the subtype is determined to be `command:move`. While `move-01` is a predicate shared among other subtypes, namely `assert:move` and `request:instruct`, the pipeline only searches for subtypes that belong to the classified speech act category.

Tense Classifier. A tense classifier determines if the original natural language utterance pertains to a *past*, *present*, or a *future* action. While it seems reasonable to use a pre-trained classifier for this three-label classification task, we built our own classifier to handle challenging cases found in our particular domain. For instance, a common shorthand command for taking a picture is *"image"*. Our framework labels the `send-image` action embedded in this command as future, but this word is not inherently associated with the future tense, nor is there any morphological information that would signal this. We implemented the same classifier from the scikit-learn library, using unigrams as features (Pedregosa et al., 2011).[5]

Rule-Based Slot Filling. The rule-based portion of this pipeline relies on regex to find and extract portions of the original AMR to fill the appropriate slots in the in-domain template. For example, for the input utterance *Move to the wall* in Fig. 2, the `command:move` template is triggered with the relation `command-02` anchoring the template and the relation `go-02` capturing the impelled action, `ARG2`, of `command-02` (Fig. 2(b)). In our restricted domain, the `ARG0` *agent* slot of `command-02` is fixed as the *Commander*, (the human instructing the robot) and `ARG1` *entity-commanded* as the *Robot* (Fig. 2(c)). The `ARG0` (mover) and `ARG1` (moved) for `move-01` in the original graph (here, *you*) is converted into the `ARG0` self-directed mover of `go-02`, and this slot is reassigned to *Robot* (Fig. 2(c)). The system then looks for the required *end point* `ARG4` slot in the original AMR, *door* in this case. The precise rules vary depending upon the template triggered, as well as the original verbal predicate used.

Aspect. Finally, we used rule-based methods for capturing aspectual information, as aspectual annotations following Donatelli et al. 2018 revealed consistent patterns associated with speech acts and subtypes. Commands consistently contain the `:completable±` annotation indicating if the commanded action is goal-oriented, which is required for execution in our problem domain (a low-bandwidth environment in which lag time in communications is expected, such that all commands require a clear endpoint in advance of execution). From this pattern, we created a rule that if an argument conveying the end point was present in the AMR, then the AMR was given a `:completable +` annotation. For common `move` and `turn` commands, the end point can be realized as the `:extent` slot (e.g., *move forward **five feet***), or the `:destination` slot (e.g., *move **to the wall***). Other speech act types present more

[5]We explored both word and character n-grams; while character n-grams can capture morphological information that signals tense (e.g., *-ed* and *-ing*), data sparsity was an issue. Unigrams, again, proved to be the most effective method.

nuanced patterns and require using speech act and tense information together. For example, assertions contain the `:ongoing - :complete +` aspectual labels within *past* tense.

Results. We evaluated the overall graph-to-graph transformation output against the 290 gold-standard in-domain graphs including all speech act categories, tense and aspect information. The *Smatch* score for this task is **F-score: .78**.[6] This system performs especially well on the `command`, `assertion`, and `express` categories, where the language tends to be predictable within this domain. Sources of errors either stem from speech act misclassification or from the rule-based methods failing to capture language variety. Misclassification of speech act and subtype can lead to more downstream errors since these elements trigger the template. Questions and requests, in particular, prove to be challenging to classify as the language of these categories are more varied. For example, *Can you describe it another way?* could be seen as a polite command, a request, or a question even to human annotators; thus, we are also evaluating the quality of the speech act distinctions. We present the results of the classifier performance using 10-fold cross-validation in Table 1.

Speech Act	Precision	Recall	F-1
Assert	.96	.96	.96
Command	.98	.94	.96
Question	.69	.81	.71
Request	.70	.92	.76
Express	.94	.83	.86
		Accuracy:	.94

Table 1: Speech act classifier performance

Other misclassification results from commands that strayed from expected language. This mainly includes statements of location (e.g., *the cleaning room*), which function as implicit movement commands in our domain. Finally, the system failed to capture certain root action predicates in the original AMRs as they were overlooked and not included in our rule-based methods—a dictionary that signals speech act subtypes.

Conclusions & **Future Work.** This paper introduces a novel yet simple approach to AMR graph-to-graph transformation, in which parser-output AMRs are converted to augmented AMRs specific to human-robot dialogue and search-and-navigation tasks. Preliminary results are quite promising, reflected by high F-1 and *Smatch* scores. However, we have yet to see this translate into performance in the end-to-end system we are working to implement. Future work will address handling truly ambiguous speech acts that cannot be determined from the language alone, which we hope to resolve by leveraging dialogue context and computer vision.

[6]Our f-score is high when compared to another AMR graph transformation task (Liu et al., 2015), but, to our knowledge, there is no directly comparable task.

References

J. L. Austin. 1975. *How to Do Things with Words*, 2nd edition. Harvard University Press and Oxford University Press.

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proc. LAW*, pages 178–186.

E. M. Bender, D. Flickinger, S. Oepen, W. Packard, and A. Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proc. of ICCS*, pages 239–249.

C. Bonial, L. Donatelli, S. M. Lukin, S. Tratz, R. Artstein, D. Traum, and C. Voss. 2019. Augmenting abstract meaning representation for human-robot dialogue. In *Proc. of DMR*, pages 199–210.

H. Bunt, J. Alexandersson, J. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proc. of LREC*, pages 430–437.

S. Cai and K. Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of ACL*, volume 2, pages 748–752.

L. Donatelli, M. Regan, W. Croft, and N. Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proc. of LAW-MWE-CxG*, pages 96–108.

M. Lindemann, J. Groschwitz, and A. Koller. 2019. Compositional semantic parsing across graphbanks. In *Proc. of ACL*, pages 4576–4585.

F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proc. of NAACL-HLT*.

M. Marge, C. Bonial, B. Byrne, T. Cassidy, W. Evans, S. Hill, and C. Voss. 2016. Applying the wizard-of-oz technique to multimodal human-robot dialogue. In *Proc. of RO-MAN*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, and Grisel O. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

J. R. Searle. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge University Press.

Frequency Matching Behavior in On-line MaxEnt Learners - Charlie O'Hara, USC

Overview Studies have repeatedly shown that language users seem to apply processes to nonce forms at a similar rate as what is observed in the lexicon as a whole (Zuraw, 2000; Ernestus & Baayen, 2003; Hayes *et al.* , 2009; Linzen *et al.* , 2013; Moore-Cantwell, 2016; Zymet, 2018b; Hughto *et al.* , 2019) Capturing both the statistical generalizations across the whole lexicon and for individual lexical items is a challenge for MaxEnt models of phonological learning, which should be able to mimic the behavior of language users. The major challenge for learners is called THE GRAMMAR-LEXICON BALANCING PROBLEM by Zymet (2018b,a)—if the lexical constraints are too active in the grammar compared to the more general grammatical constraints, the statistical generalization across the lexicon is not captured. Most work attempting to capture such biases use batch learning algorithms, directly minimizing an objective function that balances the likelihood of capturing the training data, and a prior that limits the movement of each constraint. However, on-line error-driven learners innately exhibit a bias towards limited movement of each constraint, without any explicit prior placed on the constraint weights. I use an on-line learner to examine how the innate bias of online learners can affect the grammar-lexicon balancing problem. I find that the larger the lexicon, the closer the learner matches nonce-word frequencies to the general lexical patterns.

Background Frequency matching behaviors have been observed in experiments in a variety of languages and contexts: ranging from Tagalog nasal substitution (Zuraw, 2000), voicing alternations in Dutch (Ernestus & Baayen, 2003), to Hungarian vowel harmony (Hayes *et al.* , 2009). Several proposals have attempted to model frequency matching behaviors with MaxEnt models. Moore-Cantwell & Pater (2016) use an L2 prior on the constraint weights, and approach human behavior. Zymet (2018b) and Hughto *et al.* (2019) show that the L2 prior can make the lexical constraints too active to capture the nonce-word generalizations. Zymet (2018b) and Hughto *et al.* (2019) propose different mechanisms for solving the grammar-lexicon balancing problem, but both involve an overt prior preventing the lexical constraints from receiving too much weight. The majority of this work makes use of batch learners, however Smith & Moore-Cantwell (2017) show that an on-line learner with induced (and decaying) UR constraints performs better than batch learners at capturing allomorphy in English comparatives.

The Model The simulations here use a MaxEnt grammar with two general constraints, as well as indexed variants of both general constraints for each lexical item. These lexically indexed constraints are equivalent to the lexical scales used by Hughto *et al.* (2019), and a special case of additive scaled constraints generally (Hsu & Jesney, to appear). All constraints are limited to non-negative weights.

(1)

VC_i	MAX	NOCODA	MAX_i	$NOCODA_i$
a. VC		-1		-1
b. V	-1		-1	

I use the Perceptron learning algorithm (Rosenblatt, 1958; Boersma & Pater, 2016). On each iteration of the learning algorithm, a random lexical item is sampled from the lexicon. Output forms for that item is sampled from both the target grammar, and the learner's current grammar. These two forms are compared, if they differ, the constraints violated by the learner's incorrect output are decreased, and the constraints violated by the target grammar's output are increased. In the simulations every time a mismatch occurs between the learner and the target grammar, the two general constraints MAX and NOCODA are updated (in opposite directions); but any lexically specific constraint, say MAX_i would only be updated when an error occurred on the relevant lexical item.

Simulations To evaluate whether the learner frequency matches, I compare the rate of deletion of nonce forms to the rate of deletion averaged across all lexical forms after the learner has been exposed to a fixed amount of data.

Following Hughto *et al.* (2019), I tested several distinct types of target patterns. In all of the simulations in this paper, learners were trained on data that had at most two classes of lexical items that had the same rate of variation, presented in the table in (2). In these simulations, 60% of the items maintained their final consonants at the rates in the First Portion column, and the remaining 40% maintained their final consonants at the rates in the Second Portion column.

(2)

Pattern	First Portion	Second Portion	Target	Nonce-Rate (50 items)	Learner Average
a. Categorical	1.0		1.00	1.00	1.00
b. Variable	0.7		0.7	0.715	0.723
c. Propensity	0.7	0.3	0.54	0.509	0.559
d. Variable-Lexical	0.3	1.0	0.58	0.676	0.565
e. Lexical	1.0	0.0	0.60	0.651	0.600

I ran simulations for each condition twenty times, starting with general markedness constraints (NoCoda) weighted at 50, and all other constraints weighted at zero, following (Tesar & Smolensky, 2000; Jesney & Tessier, 2011). Each simulation here had 50 items in the lexicon, and ran for 50,000 iterations with a learning rate of 0.1. With 50 items in the lexicon, the learner closely matches the average probability of coda consonant maintenance in the lexicon in the first three patterns, and overshoots the lexical generalization in patterns d and e, as shown in (2).

Impact of Lexicon Size To see the influence of lexicon size on the grammar-lexicon balance problem, I reran these simulations using a variety of different lexicon sizes, running each simulation for 1000 iterations per lexical item in the lexicon. Figure 1 shows that for the first three patterns, the discrepancy between the nonce-form deletion rate (solid line) and average deletion rate (dashed line) decreases monotonically as the number of lexical items increases. To see why this is, note that frequency matching occurs when the contribution of the specific constraints is minimal compared to the general constraints. The contribution of these specific constraints is dependent on how often the constraints update, and thus how often the learner observes an error on a specific lexical item. The more often the learner deletes a coda on a specific lexical item when the teacher produces it, the higher weighted Max$_i$ will be. Because learners start with markedness constraints weighted high, they will very often see deletion errors in early learning, as the general Max constraint approaches NoCoda. If the lexicon is small, the same lexical items will be chosen often and the specific constraints for those items will get too highly weighted; but if the lexicon is large, any single lexical item is unlikely to be selected too often, so most of the general phonotactic pattern is learned via updating the general constraints.

Fig 1: Frequency Matching with Larger Lexicons

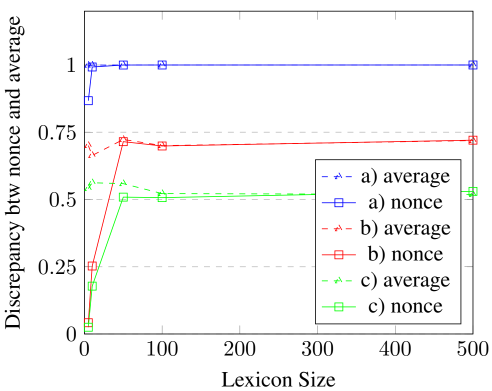


Fig 2: Overshoot with Larger Lexicons

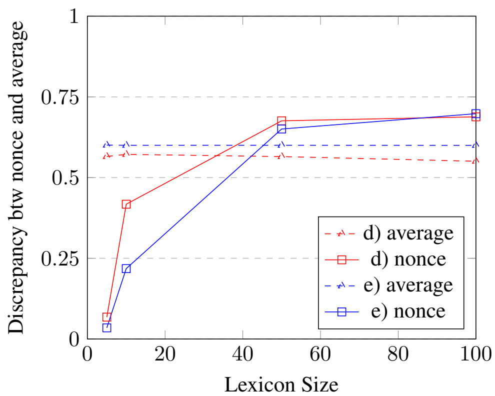


Fig 2 shows that when trained on patterns d and e, the learner overshoots the target pattern. This result resembles a bias observed by Hughto *et al.* (2019) §4.2. Further simulations showed that this overshoot is caused when one lexical class is close to, or fully categorical. In these cases, the learner learns a nonce-rate of deletion that is slightly closer to the larger categorical class's frequency than the average as a whole.

This overshoot is caused by the fact that MaxEnt grammars can never return a completely categorical mapping. The general constraints in these simulations are subject to a tug-of-war between the two lexical classes—once the general constraints have gotten close to the average mapping, the lexical idiosyncrasies must be learned. Then, if a lexical item from the class with a lower rate of deletion is sampled, Max and the relevant specific Max$_i$ are increased (and the NoCoda constraints are decreased). Then, when a lexical item from the class with a greater rate of deletion is sampled, the general constraints shift back, and the relevant specific NoCoda$_j$ is increased. If one class is smaller, it's rate of deletion will be further from the average, so each time one of its forms are sampled, it will be more likely to cause an error. Most updates on a specific lexical item will be in one direction, but if that item is variable it is possible that the learner observes updates the opposite direction, either by chance, or because the learner overshot the correct rate of deletion for that item. These updates in the opposite direction help ensure that as more of the lexical forms are learned, the amount they pull on the general constraints decreases. However, if one lexical class is categorical, the learner can never overshoot the target rate of deletion for that class. There will always be a minute tug from every lexical item in the categorical class on the general constraints. With a larger lexicon, these minute tugs compound, leading to the type of overshoot seen in Figure 2.

Without any overt priors to keep the lexical constraints from capturing much weight, on-line MaxEnt learners exhibit frequency matching behavior in most conditions.

References

Boersma, Paul, & Pater, Joe. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. *In:* McCarthy, John J., & Pater, Joe (eds), *Harmonic Grammar and Harmonic Serialism*. Equinox.

Ernestus, Mirjam, & Baayen, R. Harald. 2003. Predicting the unpredictable: Interpreting neutralized segments in Dutch. *Language*, 5–38.

Hayes, Bruce, Zuraw, Kie, Siptar, Peter, & Londe, Zsuzsa. 2009. Natural and unnatural constraints in Hungarian vowel harmony. *Language*, **85**, 822–863.

Hsu, Brian, & Jesney, Karen. to appear. Scalar Positional Markedness and Faithfulness in Harmonic Grammar. *In: Proceedings of CLS 51*.

HUGHTO, CORAL, LAMONT, ANDREW, PRICKETT, BRANDON, & JAROSZ, GAJA. 2019. Learning Exceptionality and Variation with Lexically Scaled MaxEnt. *In: Proceedings of the Society for Computation in Linguistics*, vol. 2.
JESNEY, KAREN, & TESSIER, ANNE-MICHELLE. 2011. Biases in Harmonic Grammar: The road to restrictive learning. *Natural Language & Linguistic Theory*, **29**.
LINZEN, TAL, KASYANENKO, SOFYA, & GOUSKOVA, MARIA. 2013. Lexical and phonological variation in Russian preopositions. *Phonology*, **30**, 453–515.
MOORE-CANTWELL, CLAIRE. 2016. *The representation of probabilistic phonological patterns: Neurological, behavioral, and computational evidence from the English stress system.* Ph.D. thesis, University of Massachusetts Amherst.
MOORE-CANTWELL, CLAIRE, & PATER, JOE. 2016. Gradient Exceptionality in Maximum Entropy Grammar with Lexically Specific Constraints. *Catalan Journal of Linguistics*, **15**.
ROSENBLATT, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–408.
SMITH, BRIAN, & MOORE-CANTWELL, CLAIRE. 2017. Emergent idiosyncrasy in English comparatives. *In:* LAMONT, ANDREW, & TETZLOFF, KATIE (eds), *NELS 47: Proceedings of the 47th meeting of the North East Linguistic Society*.
TESAR, BRUCE, & SMOLENSKY, PAUL. 2000. *Learnability in Optimality Theory*. MIT Press.
ZURAW, KIE. 2000. *Patterned Exceptions in Phonology*. Ph.D. thesis, UCLA, Los Angeles.
ZYMET, JESSE. 2018a. Learning a Frequency-Matching Grammar together with Lexical Idiosyncrasy: MaxEnt versus Hierarchical Regression. *In:* HOUT, KATHERINE, MAI, ANNA, MCCOLLUM, ADAM, ROSE, SHARON, & ZASLANSKY, MATT (eds), *Proceedings of the 2018 Annual Meeting on Phonology*. Washington, DC: Linguistics Society of America.
ZYMET, JESSE. 2018b. *Lexical propensities in phonology: corpus and experimental evidence, grammar, and learning*. Ph.D. thesis, University of California Los Angeles.

Frequency-(in)dependent regularization in language production and cultural transmission

Emily Morgan (UC Davis; eimorgan@ucdavis.edu) and Roger Levy (MIT; rplevy@mit.edu)

In cases of variation in language, how do people learn and reproduce probabilistic distributions over linguistic forms? Given a certain amount of variation in their linguistic input, speakers could aim to reproduce the variation exactly (i.e. to *probability match*) or could instead aim to *regularize*—to make their productions more consistent by reproducing the most frequent variant *even more frequently* than it was heard in the input. While we know that people retain detailed statistics about their linguistic input (Levy, 2008; Arnon & Snider, 2010), there is also evidence for regularization in language learning (Hudson Kam & Newport, 2005; Reali & Griffiths, 2009), although the circumstances that lead to regularization versus probability matching are not yet well understood. Morgan and Levy (2015) found evidence in corpus data that *binomial expressions* of the form "X and Y" are more regularized the higher their frequency—i.e. their ordering preferences (e.g. "bread and butter" vs. "butter and bread") are more extreme when the two words ("bread" and "butter") co-occur in a binomial more frequently, regardless of order. This finding is puzzling because previous experimental research does not suggest that regularization should be frequency-dependent. However, when we find systematic patterns in corpus data, we would like to be able to attribute them to motivated preferences (based on language learning and/or production; Hawkins, 2004). Does this corpus data in fact provide evidence for regularization in online language processing, and if so, does speakers' regularization behavior depend on an item's frequency, contrary to previous claims?

We demonstrate that frequency-dependent regularization can arise diachronically through a combination of a frequency-independent synchronic regularization bias and the bottleneck effect of cultural transmission. We simulate diachronic language change using an Iterated Learning Model (Smith, 2009) in which speakers in successive generations iteratively learn binomial expression preferences from the previous generations' productions and then generate their own productions. We augment the standard model with a regularization bias that applies during production. Although the bias itself is frequency-*independent*, we demonstrate that frequency-*dependent* regularization emerges from the iterated learning process. For lower frequency items, a tighter bottleneck (fewer productions per generation) favors convergence to the prior. Because prior preferences depend only on the words in the binomial—not on its frequency—the bottleneck thus prevents the regularization bias from having a strong effect. With increasing frequency, a wider bottleneck (more productions per generation) increasingly transmits the effects of the regularization bias across generations. Our model thus correctly predicts the qualitative pattern of frequency-dependent regularization (Fig 1).

Moreover, our model correctly predicts the observed language-wide distribution of ordering preferences in Morgan and Levy's (2015) binomial corpus. For each binomial expression in the corpus, we predict its ordering preference based on its frequency of occurrence (as well as other word-level properties). Our model correctly predicts the multimodal distribution of ordering preferences found in the corpus (Fig 2).

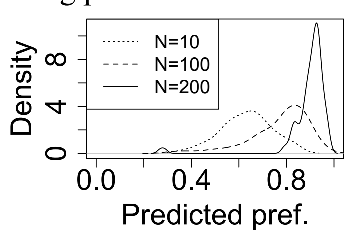


Fig 1. Model-predicted preferences for a hypothetical binomial, from 0 (always one order) to 1 (always the other), are more extreme (closer to 0 and/or 1) with increasing number of productions per generation N.

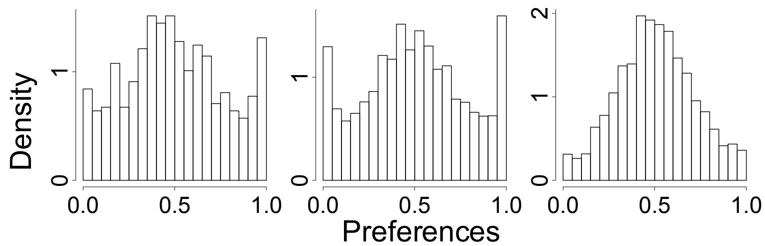


Fig 2. Language-wide distribution of preferences in corpus data (left) and as predicted by a model with (center) and without (right) a regularization bias.

Our model thus confirms previous demonstrations of a regularization bias in language learning and/or production, but demonstrates that frequency-dependent regularization in a corpus distribution does not imply that frequency influences regularization at the level of individual speakers. Rather, the pattern of frequency-dependent regularization seen in corpus data can arise from the interaction of a frequency-independent bias in online language processing and the bottleneck effect of cultural transmission.

We conclude by questioning why language learning and/or production might include a regularization bias. One hypothesis is that an online regularization bias promotes efficiency in language processing by reducing the choices that must be made, hence reducing the cost of online utterance planning. Another hypothesis relates to difficulty during early learning rather than during online production. Focusing on one variant during learning may reduce cognitive load, and therefore regularization may be both particularly prevalent and particularly beneficial during early language learning when cognitive resources are more limited than in adulthood.

This work was previously published in: Morgan, E., & Levy, R. (2016). Frequency-Dependent Regularization in Iterated Learning. In S. G. Roberts, C. Cuskley, L. McCrohon, L. Barceló-Coblijn, O. Fehér, & T. Verhoef (Eds.), *The Evolution of Language: Proceedings of the 11th International Conference (EVOLANG 11)*. http://evolang.org/neworleans/papers/193.html

References

Arnon, I., & Snider, N. (2010). More than words: Frequency effects for multi-word phrases. *Journal of Memory and Language*, *62*(1), 67–82.

Hawkins, J. A. (2004). *Efficiency and Complexity in Grammars*. Oxford University Press.

Hudson Kam, C. L., & Newport, E. L. (2005). Regularizing Unpredictable Variation: The Roles of Adult and Child Learners in Language Formation and Change. *Language Learning and Development*, *1*(2), 151–195.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, *106*(3), 1126–1177.

Morgan, E., & Levy, R. (2015). Modeling idiosyncratic preferences: How generative knowledge and expression frequency jointly determine language structure. In D. C. Noelle, R. Dale, A. S. Warlaumont, J. Yoshimi, T. Matlock, C. D. Jennings, & P. P. Maglio (Eds.), *37th Annual Meeting of the Cognitive Science Society* (pp. 1649–1654). Austin, TX: Cognitive Science Society.

Reali, F., & Griffiths, T. L. (2009). The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition*, 111(3), 317–328.

Smith, K. (2009). Iterated learning in populations of Bayesian agents. *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, 697–702.

Crosslinguistic Word Orders Enable an Efficient Tradeoff of Memory and Surprisal (Abstract)

Michael Hahn
Stanford University
mhahn2@stanford.edu

Richard Futrell
University of California, Irvine
rfutrell@uci.edu

Memory limitations are well-established as a factor in human online sentence processing (Gibson, 1998; Lewis and Vasishth, 2005), and have been argued to account for crosslinguistic word order regularities. For example, the Performance–Grammar Correspondence Hypothesis of Hawkins (1994) holds that forms which are easier to produce and comprehend end up becoming part of the grammars of languages. We build on expectation-based models of language processing (Levy, 2008) and on the theory of lossy compression (Cover and Thomas, 2006) to develop a highly general information-theoretic notion of memory efficiency in language processing, in terms of a trade-off of surprisal and memory usage. We derive a method for estimating a lower bound on the memory efficiency of languages from corpora, and apply our method to corpora from 54 languages to test the idea that word order is structured to reduce processing effort under memory limitations. We find that word orders tend to support efficient tradeoffs between memory and surprisal, suggesting that word order rules are structured to enable efficient online processing.

Background Surprisal theory (Levy, 2008) posits that the processing effort on a word w_t in context $w_1 \dots w_{t-1}$ is proportional to the **surprisal** of the word in context:

$$S = -\log \mathrm{P}(w_t | w_1 \dots w_{t-1}). \quad (1)$$

Experimental work has confirmed that surprisal is a reliable and linear predictor of processing effort as reflected in reading times (Smith and Levy, 2013).

However, surprisal theory as presented above cannot in principle account for effects of memory limitations on online processing, because Equation 1 represents surprisal as experienced by an idealized listener who accurately remembers the entire history of previous words $w_{1\dots t-1}$. More realistically, human listeners deploy memory resources that maintain imperfect representations of the preceding context (Lewis and Vasishth, 2005; Futrell and Levy, 2017). If m_t is a listener's memory state after hearing $w_1 \dots w_{t-1}$, then the true surprisal experienced by the listener will be:

$$S_M := -\log_2 \mathrm{P}(w_t | m_t), \quad (2)$$

which must be larger than Eq. 1 on average (Cover and Thomas, 2006).

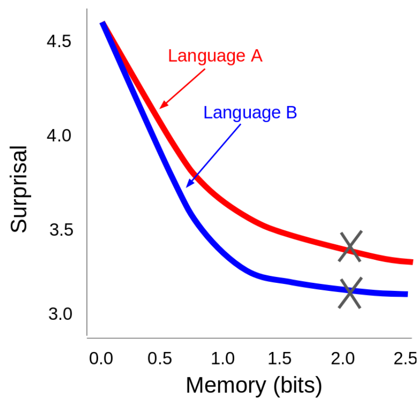


Figure 1: Conceptual tradeoff between memory and surprisal for two languages. In Language A (blue), a listener storing 1 bit can achieve average surprisal 3.5, while the same level of surprisal requires 2 bits of memory for a listener in Language B (red).

Memory–surprisal tradeoff. These considerations imply a *tradeoff between memory and surprisal*: A listener maintaining higher-precision memory representations m_t will, on average, incur lower surprisal, at the cost of higher memory load. The idea of the memory-surprisal tradeoff is visualized in Fig. 1: for each desired level of average surprisal, there is a minimum number of bits of information which must be stored about context. The shape of the trade-off is determined by the language, and in particular its word order: some languages enable more efficient trade-offs than others by forcing a listener to store more bits in memory to achieve the same level of average

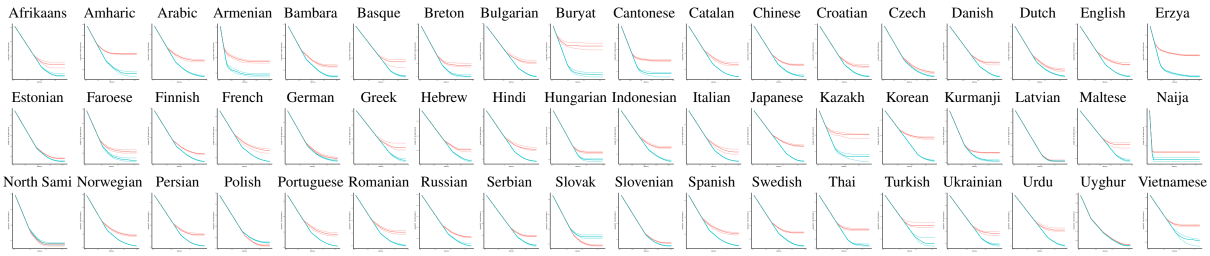


Figure 2: Tradeoffs between memory (x axis) and surprisal (y axis) in 54 languages, for real orderings (blue) and counterfactual baseline grammars (red). We provide 95% confidence bands for different model runs on the real languages, and for the median across different baseline grammars.

surprisal.

Theoretical Results In Theorem 1 below, we derive a bound on the memory-surprisal tradeoff curve which can be easily estimated from corpora. Let I_t be the conditional mutual information between words that are t steps apart, conditioned on the intervening words:

$$\mathrm{I}_t := \mathrm{I}[w_t, w_0 | w_{1\ldots t-1}].$$

This quantity measures how much predictive information the word t steps in the past contains about the current word.

Theorem 1. *Let T be a positive integer, and consider a listener using at most $\sum_{t=1}^{T} t\,\mathrm{I}_t$ bits of memory on average. Then this listener will incur average surprisal at least $\mathrm{H}[w_t|w_{<t}] + \sum_{t>T} \mathrm{I}_t$.*

The theorem allows us to estimate the extra surprisal associated with each amount of memory capacity for a language. The quantities I_t can be estimated as the difference between the cross-entropy of language models that have access to the last $t-1$ or t words. Given such estimates of I_t, we estimate tradeoff curves as in Figure 1 by tracing out $T = 1, 2, \ldots$.

Experimental Results We tested whether word orders as found in natural language grammars provide efficient memory-surprisal tradeoffs. To this end, we compared corpora of real languages against hypothetical reorderings of those languages under random baseline grammars. We used treebanks of 54 languages from Universal Dependencies 2.3 (Nivre et al., 2018).

For each language, we constructed counterfactual word order rules by adapting the methodology of Gildea and Temperley (2010) to Universal Dependencies: For each syntactic relation (subject, object, ...) used in the treebank annotation, we randomly sampled its position relative to the head and other of siblings. For each language and each such set of rules, we reordered the treebank according to these counterfactual word order rules.

For each language and its counterfactually ordered versions, we estimated the memory-surprisal tradeoff (Theorem 1) using an LSTM recurrent neural language model, considering all integers $T = 1, \ldots, 20$. Hyperparameters were tuned, for each language, to minimize average cross-entropy on counterfactual versions, introducing a conservative bias against our hypothesis.

Tradeoff curves are shown in Figure 2. In 50 out of 54 languages, the observed orderings led to more favorable tradeoffs than 50% of the counterfactual orderings ($p < 0.0001$; Exceptions: Latvian, North Sami, Polish, and Slovak).

Taken together, our results suggest that, across languages, word order in part reflects pressures towards efficient online processing under memory limitations.

References

Thomas M. Cover and J.A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ.

R Futrell and R Levy. 2017. Noisy-context surprisal as a human sentence processing cost model. In *EACL*.

E. Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

D Gildea and D Temperley. 2010. Do Grammars Minimize Dependency Length? *Cognitive Science*.

JA Hawkins. 1994. *A performance theory of order and constituency*.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

R Lewis and S Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*.

Joakim Nivre et al. 2018. Universal dependencies 2.3.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Reconciling Historical Data and Modern Computational Models in Corpus Creation

Joseph Rhyne
Department of Linguistics
Cornell University
jtr92@cornell.edu

1 Overview

We live in a time of unprecedented access to linguistic data, from audio recordings to corpora of billions of words. Linguists have used these resources to advance their research and understanding of language. Historical linguistics, despite being the oldest linguistic subfield, has lagged behind in this regard. However, this is due to several unique challenges that face the subfield. Historical data is plagued by two problems: a lack of overall data due to the ravages of time and a lack of model-ready data that have gone through standard NLP processing. Barring the discovery of more texts, the former issue cannot be solved; the latter can, though it is time-consuming and resource-intensive. These problems have only begun to be addressed for well-documented language families like Indo-European, but even within these progress is slow.

There have been numerous advances in synchronic models for basic NLP tasks like POS and morphological tagging. However, modern models are not designed to work with historical data: they depend on large volumes of data and pre-tagged training sets that are not available for the majority of historical languages. Some have found success with methods that are designed to imitate traditional historical approaches, e.g. (Bouchard-Côté et al., 2013; McMahon and McMahon, 2003; Nakleh et al., 2005), but, if we intend to use state-of-the-art computational tools, they are essentially incompatible. This is an important challenge that computational historical linguists must address if they are going to meet the standards set by both modern corpora and historical analyses. This paper approaches the issue by treating historical data in the same way as a low-resource language (Fang and Cohn, 2017; Buys and Botha, 2016; Mishra et al., 2018) and integrating data from modern descendant languages. Through these approaches, we are able to tag a number of new texts in Old Slavic languages for part-of-speech. Many of these texts have never previously been tagged. With these problems overcome, we can create new corpora of historical language and thus dramatically increase both the number and type of diachronic linguistic investigations.

2 Modern approaches to historical data

Historical Data as low-resource language. This challenge is not unique to historical data. Thousands of languages across the world also lack the necessary resources for standard computational analyses and models. These low-resource languages have not been sufficiently documented and thus do not have adequate datasets for model-training. Many different approaches have been proposed on how to deal with this issue for low-resource languages. For example, (Buys and Botha, 2016) improve results through the use of parallel forpora, which could be helpful for those languages that have modern high-resource language translations. Others have proposed feature projection (Mishra et al., 2018) for morphologically-complex languages. In this paper, we exploit the approach called *Model Transfer* (Fang and Cohn, 2017). Here, a bilingual dictionary, monolingual corpora in both the high- and low-resource languages, and a small annotated corpus for the low-resource language, are used to train a model through joint training from both sources. The bilingual dictionary and monolingual corpora are used to train cross-lingual word embeddings, while language-dependent information can be learned from the small annotated corpus. The lack of available dictionaries for some languages is a pitfall for Model Transfer.

Extending modern language data. Historical data does not exist within a vacuum. One avenue that we could exploit is its relationship to descendant and related languages, i.e. how Modern English is a descendant of Middle English. We might leverage the large amount of pre-processed data available for the modern languages to help create the models for their older stages. We call this *Model Extension*, where a model is created to tag one language using training data from a related language. In this paper, we train models on modern data and use them to tag the older texts. Thus the model is extending to a new linguistic domain. No matter the approach, manual annotation is an option, and it goes a long way in helping to train models on these limited data.

3 Data

For this paper, we experiment on Old Slavic languages, focusing on Old Church Slavonic (OCS; 46 texts: 10 tagged, 36 untagged), Old East Slavic (OES; 35 texts: 32 tagged, 3 untagged), and Old Polish (OP; 20 untagged texts). These are good candidates because there are (1) resources for some of the languages (OCS and OES) and (2) well-documented modern descendant languages, i.e. Bulgarian for OCS, Russian for OES, and Polish for OP. Some pre-tagged texts for OCS and OES were taken from the TOROT treebank (Eckhoff and Berdiceviskis) to be used as training and test data. Untagged texts in all three languages were taken from sites like Thesaurus Indogermanischer Text- und Sprachmaterialien. OCS was the only language for which an extensive dictionary could be found, thus it is the only language to use *Model Transfer*. Word-embeddings were trained for the languages using the gathered texts. Models for the modern language were trained using data taken from Universal Dependencies.

4 Models

In order to tag the corpus we used an extension of a sequence tagging network, based on (Reimers and Gurevych, 2017) and (Arakelyan et al., 2018). These are based on BiLSTM networks from (Huang et al., 2015). For the models, we use a variety of both pre-trained embeddings for modern languages and newly-trained embeddings for the old languages, using Word2Vec (Mikolov et al., 2013). This set-up of the network can be seen in Figure 1.

Based on this architecture, we trained three types of models: (1) Normal Models using the pre-tagged data for OCS and OES, (2) Model Transfer for OCS using an OCS-English dictionary and the British National Corpus, and (3) Modern Model Extensions using Universal Dependency models for Buglarian, Russian, and Polish. The OCS Model Transfer model had an additional requirement: following (Fang and Cohn, 2017), after the input of raw text, we use cross-lingual word embeddings (Ammar et al., 2016) instead of the usual

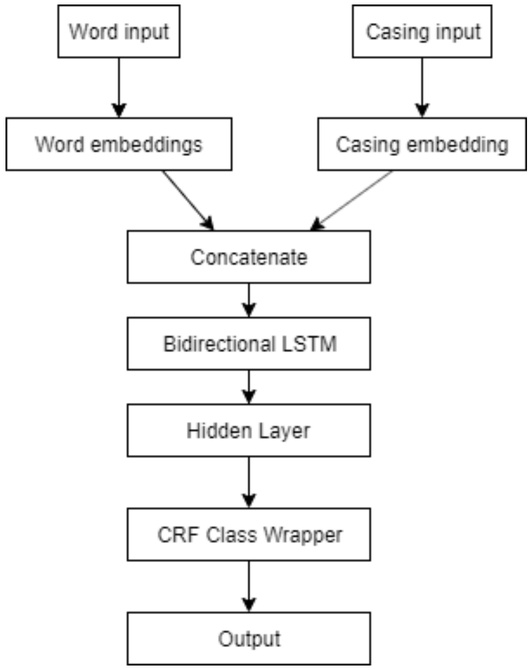


Figure 1: Basic architecture, showing the layers of the network used to create the models

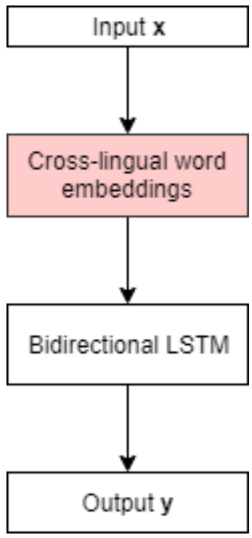


Figure 2: OCS Model Transfer has only one difference: the use of cross-lingual word embeddings (Ammar et al., 2016)

Language	*Normal*	*Model Transfer*	*Modern Extensions*	*Universal Dependency*
OCS	75.63	76.54	65.23	87.40
OES	69.60	N/A	70.95	83.91
Old Polish	N/A	N/A	69.82	84.64

Table 1: Accuracies for test set tagging in each language across different models

monolingual word embeddings. These combine monolingual embeddings trained using *word2vec* by projecting them onto a common space, which is learned through the bilingual dictionary. This is then used with the large tagged corpus of the high-resource language in the training, to then be applied to the untagged historical data. The tagging model itself uses the same BiLSTM architecture described above. The Model Transfer workflow can be seen in Figure 2. For comparison, Universal Dependency models were trained using the UD data for the three modern related languages: Bulgarian, Russian, and Polish.

5 Results

All models were subject to the same test set in each of the languages. Because there was no previously tagged corpus, the test set for Old Polish was hand-tagged for this project. This determined their POS tagging accuracies, which are compiled in Table 1.

None of the models achieved the same level of accuracy seen by the modern Universal Dependency models. The normally-trained models for OCS and OES were close, as a result of their pre-tagged data. In general, we can see that the use of Model Transfer and Model Extension does not negatively impact the POS tagging accuracy. The Extension model for OCS is lower than for the others, but this is likely due to dramatic morphological differences between OCS and its modern relative Bulgarian. While the overall accuracies are not as high as most modern language models, they are not so low as to be discouraging. They do show that, in the instance of a language like Old Polish, Model Transfer and Extension are serviceable methods for tagging new texts. Even at a 70% POS-tagging accuracy, these methods provide a great first-pass run in the pipeline of corpus creation for a language without resources. Moreover, this maintenance of a comparably high accuracy shows that we can leverage different stages of a language to fill in gaps in our models. This is still likely dependent on other diachronic factors, e.g. we might expect a lower accuracy for an older morphologically-complex language when its descendant form is much more morphologically-simple.

6 Conclusion

The results so far do not meet the standards set by modern models, but they do still serve as a good first-pass run that can be improved with manual annotation and other tagging methods. This will still save valuable time and increase the number and type of resources available to historical linguistics. This in turn will further aid historical linguists in both their diachronic and synchronic analyses for the languages and language families included in the new corpora, e.g. (Rhyne, Forthcoming). This can only improve with access to more data. Nevertheless, it is still promising that models can be extended relatively well from modern languages to their ancestors. Moreover, there are still multiple low-resource language approaches that can still be used, such as parallel corpora (Buys and Botha, 2016). This would be especially useful for languages that have extensive English or other modern translations. We might also try to use dictionaries of modern descendant languages in our Model Transfer approach.

Thus, this paper attempts to fill in a gap that continues to plague historical linguistics. The results are still lacking, but they show signs of improvement. With more time and resources, other methods could be explored, particularly those that depend on extensive pre-tagged data. Nevertheless, through efforts like these, we can improve the quality of data within historical linguistics, making it more approachable to all linguists and matching the standards already established in the rest of the field.

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith.

2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

Gor Arakelyan, Karen Hambardzumyan, and Hrant Khachatrian. 2018. Towards JointUD: Part-of-speech tagging and lemmatization using recurrent neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 180–186, Brussels, Belgium. Association for Computational Linguistics.

Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, , and Dan Klein. 2013. Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 110:4224–4229.

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. *CoRR*, abs/1606.04279.

Hanne Martin Eckhoff and Aleksandrs Berdiceviskis. Linguistics vs. digital editions: The Tromsø Old Russian and OCS Treebank. In *Scripta and e-Scripta 14-15*, pages 9–25.

Meng Fang and Trevor Cohn. 2017. Model transfer for tagging low-resource languages using a bilingual dictionary. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 587–593, Vancouver, Canada. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

April McMahon and Robert McMahon. 2003. Finding families: quantitative methods in language classification. *Transactions of the Philological Society*, 101(1):7–55.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Pruthwik Mishra, Vandan Mujadia, and Dipti Sharma. 2018. Pos tagging for resource poor indian languages through feature projection.

Luay Nakleh, Tandy Warnow, Don Ringe, and Steven N. Evans. 2005. A comparison of phylogenetic reconstruction methods on an indo-european dataset. *Transactions of the Philological Society*, 103:171–192.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Joseph Rhyne. Forthcoming. Contrasts in case usage under negation in old church slavonic. In *Proceedings of the 29th Annual UCLA Indo-European Conference*, Bremen. Hempen.

Tensor Product Decomposition Networks: Uncovering Representations of Structure Learned by Neural Networks

R. Thomas McCoy,[1] Tal Linzen,[1] Ewan Dunbar,[2] & Paul Smolensky[3,1]
[1]Department of Cognitive Science, Johns Hopkins University
[2]Laboratoire de Linguistique Formelle, CNRS - Université Paris Diderot - Sorbonne Paris Cité
[3]Microsoft Research AI, Redmond, WA USA
`tom.mccoy@jhu.edu`, `tal.linzen@jhu.edu`,
`ewan.dunbar@univ-paris-diderot.fr`, `smolensky@jhu.edu`

Recurrent neural networks (RNNs; Elman, 1990) use continuous vector representations, yet they perform remarkably well on tasks that depend on compositional symbolic structure, such as machine translation. The inner workings of neural networks are notoriously difficult to understand, so it is far from clear how they manage to encode such structure within their vector representations.

We hypothesize that they do this by learning to compile symbolic structures into vectors using the tensor product representation (TPR; Smolensky, 1990), a general schema for mapping symbolic structures to numerical vector representations. To test this hypothesis, we introduce Tensor Product Decomposition Networks (TPDNs), which are trained to use TPRs to approximate existing vector representations. If a TPDN is able to closely approximate the representations generated by an RNN, it would suggest that the RNN's strategy for encoding compositional structure is to implicitly implement the type of TPR used by the TPDN.

Using this method, we show that networks trained on artificial tasks using digit sequences discover structured representations appropriate to the task; e.g., a model trained to copy a sequence will encode left-to-right position (*first*, *second*, *third*...), while a model trained to reverse a sequence will use right-to-left position (*last*, *second-to-last*, *third-to-last*...). Thus, our analysis tool shows that RNNs are capable of discovering structured, symbolic representations. Surprisingly, however, we also show, in several real-world networks trained on natural language processing tasks (e.g., sentiment prediction), that the representations used by the networks show few signs of structure, being well approximated by an unstructured (bag-of-words) representation. This finding suggests that popular training tasks for sentence representation learning may not be sufficient for inducing robust structural representations.

Tensor Product Decomposition Networks: To represent a symbolic structure with a TPR, each component of the structure (e.g., each element in a sequence) is called a **filler**, and the fillers are paired with **roles** that represent their positions (Figure 2a). Each filler f_i and — crucially — each **role** r_i has a vector embedding; these two vectors are combined using their tensor product $f_i \otimes r_i$, and these tensor products are summed to produce the representation of the sequence: $\sum f_i \otimes r_i$.

To test whether a set of vector encodings can be approximated with a TPR, we introduce the Tensor Product Decomposition Network (TPDN; Figure 1c), a model that is trained to use TPRs to approximate a given set of vector representations that have been generated by an RNN encoder. Approximation quality is evaluated by feeding the outputs of the trained TPDN into the decoder from the original RNN and measuring the accuracy of the resulting hybrid architecture (Figure 1d). We refer to this metric as *substitution accuracy*.

Approximating RNN representations: To establish the effectiveness of the TPDN at uncovering the structural representations used by RNNs, we first apply the TPDN to sequence-to-sequence networks (Sutskever et al., 2014) trained on a copying objective: they are expected to encode a sequence of digits and then decode that encoding to reproduce the same sequence (Figure 1a).

We ran this experiment with two types of sequence-to-sequence RNNs: linear RNNs, which process sequences in linear order, and tree RNNs, which process sequences in accordance with a tree structure. These experiments revealed that the encodings of the linear RNN could be approximated very closely (with a substitution accuracy of over 0.99 averaged across five runs) with a TPR using the bidirectional role scheme, which encodes the distance from the start of the sequence and

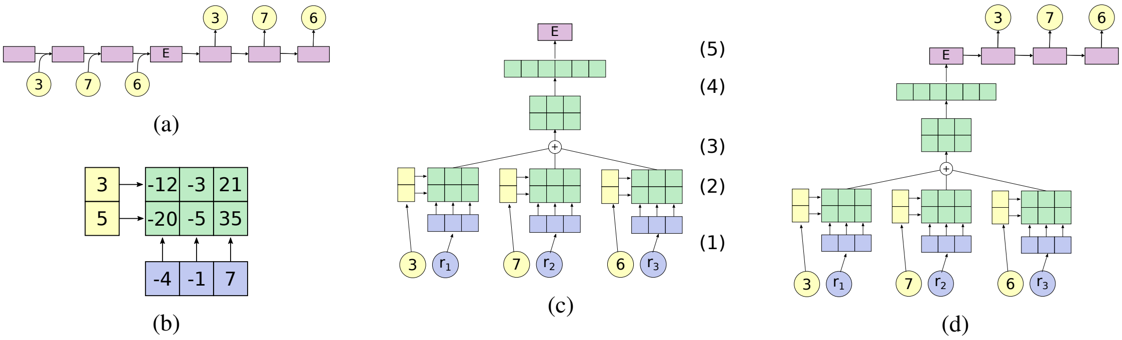


Figure 1: **(a)** A sequence-to-sequence model performing copying. **(b)** The tensor product. **(c)** A TPDN trained to approximate the encoding E from Figure 1a: (1) The fillers and roles are embedded. (2) The fillers and roles are bound together using the tensor product. (3) The tensor products are summed. (4) The sum is flattened into a vector by concatenating the rows. (5) A linear transformation is applied to get the final encoding. **(d)** The architecture for evaluation: using the original sequence-to-sequence model's decoder with the trained TPDN as the encoder.

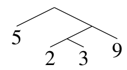
the distance from the end of the sequence. By contrast, the tree RNN was closely approximated by a role scheme encoding tree position but not by any of the role schemes encoding linear position. These results show that RNNs are capable of learning to generate compositional symbolic representations and that the nature of these representations is closely related to the RNN's structure.

Approximating sentence representations: We now investigate whether the TPDN's success with digit sequences will extend to naturally occurring linguistic data. We use sentence representations from four natural language processing models: two linear RNNs, InferSent and Skip-thought; and two tree RNNs, the Stanford sentiment model (SST) and SPINN. All four models are reasonably well approximated with a bag of words, which only encodes which words are in the sentence and does not encode any sort of sentence structure; other role schemes which do encode structure showed only modest improvements (Figure 3b).

Conclusion: With heavily structure-sensitive tasks, sequence-to-sequence RNNs learned representations that were extremely well approximated by tensor-product representations. By contrast, sentence encoders from the natural language processing literature could be reasonably well-approximated with an unstructured bag of words, suggesting that the representations of these models were not very structure-sensitive. These results suggest that, when RNNs learn to encode compositional structure, they do so by adopting a strategy similar to TPRs, but that existing tasks for training sentence encoders are not sufficiently structure-sensitive to induce RNNs to encode such structure.

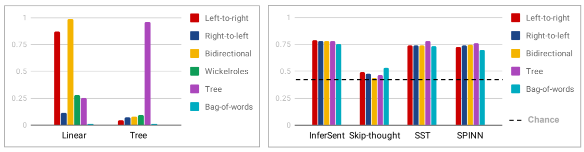
	5	2	3	9
Left-to-right	0	1	2	3
Right-to-left	3	2	1	0
Bidirectional	(0, 3)	(1, 2)	(2, 1)	(3, 0)
Wickelroles	#_2	5_3	2_9	3_#
Tree	L	RLL	RLR	RR
Bag-of-words	r_0	r_0	r_0	r_0

(a)



(b)

Figure 2: (a) The filler-role bindings assigned by the six role schemes to the sequence 5239. (b) The tree used to assign tree roles to this sequence.



(a) (b)

Figure 3: Results. (a) Substitution accuracies for linear and tree RNNs trained on copying. (b) The proportion of test examples on which classifiers trained on sentence encodings gave the same predictions for these encodings and for their TPDN approximations, averaged across four tasks. The dotted line indicates chance performance.

References

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*.

Inferring Minimalist Grammars with an SMT-Solver

Sagar Indurkhya
MIT
77 Mass. Ave.
Cambridge, MA 02139
indurks@mit.edu

Overview. The Minimalist Grammar (MG) formalism (Stabler, 1996) is a well established formal model of syntax inspired by the Minimalist Program (Chomsky, 1995). We introduce (1) a novel parser for MGs[1], encoded as a system of first-order logic formulae that may be evaluated using a solver for Satisfiability Modulo Theories (SMT) (De Moura and Bjørner, 2008; Barrett and Tinelli, 2018), and (2) a novel procedure for inferring MGs using this parser. The input to this procedure is a sequence of sentences that have been annotated with syntactic relations such as semantic role labels (connecting arguments to predicates) and subject-verb agreement. The output of this procedure is a set of MGs, each of which is able to parse the sentences in the input sequence such that the parse for a sentence has the same syntactic relations as those specified in the annotation for that sentence. We applied this procedure to a set of sentences annotated with syntactic relations and evaluated the inferred grammars using cost functions inspired by the Minimum Description Length (MDL) principle (Barron et al., 1998; Grünwald, 2007) and the Subset principle (Berwick, 1985; Wexler, 1993). Inferred grammars that were *optimal* with respect to certain combinations of these cost functions were found to align closely with contemporary theories of Minimalist syntax (Hornstein et al., 2005; Adger, 2003; Radford, 1997), producing the prescribed syntactic structures for a range of constructions that include ditransitive predicates, passivization and Wh-fronting for question formation.

Inference Procedure. Our inference procedure takes the form of a computational model of language acquisition (Chomsky, 1965; Berwick, 1985) consisting of: (1) an initial state, S_0, consisting of a system of first-order logical formulae that serve as axioms for deducing the class of minimalist lexicons; (2) the input, consisting of a sequence of n sentences, denoted $I_1, I_2, \ldots, I_n$, each of which is annotated with syntactic relations between pairs of words in the sentence; (3) a function, Q, that takes as input a state, S_i, and an annotated sentence, I_i, and outputs the successor state, S_{i+1}; (4) a function, R, that maps a state S_i to a set of MG lexicons, G_i, with the property that for each sentence I_j in the input sequence, each lexicon $L \in G_i$ can produce a parse p_j^L such that the syntactic relations in p_j^L parse match those specified in the annotation of s_j. In the case of the initial state, S_0, since there are no constraints yet imposed by the input, $R(S_0)$ will map to the set of all minimalist lexicons. The procedure consumes the input sequence one annotated sentence at a time, using Q to drive the initial state, S_0, to the final state, S_n; the function R is then applied to S_n to produce a set of MG lexicons, G_n, that constitutes the output of the inference procedure.

We implemented this inference procedure by encoding an MG parser as a system of first-order, quantifier-free logical formulas that could be solved with the Z3 SMT-solver (De Moura and Bjørner, 2011; Cadar and Sen, 2013).[2] This system of formulas is composed of formulas for MG parse trees that are connected (by way of shared symbols) to a formula for an MG lexicon (i.e. S_0); by imposing constraints on the formulas for parse trees (via Q), the set of solutions to the lexicon formula is restricted (i.e. R is constrained). When the inference procedure consumes an annotated sentence from the input sequence, the function Q: (1) instantiates a formula for an MG parse;

[1] We used the chain-based formulation of MGs presented in (Stabler and Keenan, 2003).

[2] This approach is inspired by earlier work that modeled grammar with logic (Pereira and Warren, 1983; Rayner et al., 1988; Stabler, 1993; Rogers, 1998; Graf, 2013).

I_i	Sentence	Locality Constraints
I_1	who has eaten/V icecream/N?	$\theta_{\text{eaten}}[s\colon \text{who}, o\colon \text{icecream}]$, $Agr_{\text{has}}[s\colon \text{who}]$
I_2	icecream/N was eaten/V.	$\theta_{\text{eaten}}[o\colon \text{icecream}]$, $Agr_{\text{was}}[s\colon \text{icecream}]$
I_3	who was eating/V icecream/N?	$\theta_{\text{eating}}[s\colon \text{who}, o\colon \text{icecream}]$, $Agr_{\text{was}}[s\colon \text{who}]$
I_4	was pizza/N eaten/V?	$\theta_{\text{eaten}}[o\colon \text{pizza}]$, $Agr_{\text{was}}[s\colon \text{pizza}]$
I_5	what has john/N eaten/V?	$\theta_{\text{eaten}}[s\colon \text{john}, o\colon \text{what}]$, $Agr_{\text{has}}[s\colon \text{john}]$
I_6	has mary/N eaten/V pizza/N?	$\theta_{\text{eaten}}[s\colon \text{mary}, o\colon \text{pizza}]$, $Agr_{\text{has}}[s\colon \text{mary}]$
I_7	was john/N eating/V pizza/N?	$\theta_{\text{eating}}[s\colon \text{john}, o\colon \text{pizza}]$, $Agr_{\text{was}}[s\colon \text{john}]$
I_8	what was mary/N eating/V?	$\theta_{\text{eating}}[s\colon \text{mary}, o\colon \text{what}]$, $Agr_{\text{was}}[s\colon \text{mary}]$
I_9	what was eaten/V?	$\theta_{\text{eaten}}[o\colon \text{what}]$, $Agr_{\text{was}}[s\colon \text{what}]$
I_{10}	was mary/N given/V pizza/N?	$\theta_{\text{given}}[o\colon \text{pizza}, i\colon \text{mary}]$, $Agr_{\text{was}}[s\colon \text{mary}]$
I_{11}	what has mary/N given/V john/N?	$\theta_{\text{given}}[s\colon \text{mary}, o\colon \text{what}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{mary}]$
I_{12}	mary/N has given/V john/N money/N.	$\theta_{\text{given}}[s\colon \text{mary}, o\colon \text{money}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{mary}]$
I_{13}	who was money/N given/V to/P?	$\theta_{\text{given}}[o\colon \text{money}, i\colon \text{to who}]$, $Agr_{\text{was}}[s\colon \text{money}]$
I_{14}	who has john/N given/V money/N to/P?	$\theta_{\text{given}}[s\colon \text{john}, o\colon \text{money}, i\colon \text{to who}]$, $Agr_{\text{has}}[s\colon \text{john}]$

Table 1: Model Input — A sequence of sentences annotated with syntactic relations. Some phonetic forms have their category pre-specified, indicated by a suffix of a slash followed by the category. Locality constraints include agreement (Agr) and predicate-argument structure (i.e. a θ grid), with the predicate indicated in the suffix and the subject, object and indirect object components marked by "s:", "o:" and "i:" respectively. The type of the sentence, *declarative* or *interrogative*, is indicated by the end-of-sentence punctuation.

Lexicon-A	Lexicon-B
eaten/V :: $=x_4, \sim x_4$	eaten/V :: $=x_5, \sim x_1$
eating/V :: $=x_4, \sim x_4$	eating/V :: $=x_5, \sim x_1$
given/V :: $=x_4, =x_4, \sim x_4$	given/V :: $=x_5, =x_5, \sim x_1$
given/V :: $=x_2, =x_4, \sim x_4$	has/T :: $=x_0, +l, \sim x_2$
has/T :: $=x_4, +l, \sim x_0$	icecream/N ::$\sim x_5$
has/T :: $=x_4, +l, \sim x_4$	icecream/N ::$\sim x_5, -l$
icecream/N ::$\sim x_4$	john/N ::$\sim x_5$
icecream/N ::$\sim x_4, -l, -r$	john/N ::$\sim x_5, -l$
john/N ::$\sim x_4$	mary/N ::$\sim x_5, -l$
john/N ::$\sim x_4, -l$	money/N ::$\sim x_5$
mary/N ::$\sim x_4, -l$	money/N ::$\sim x_5, -l$
mary/N ::$\sim x_4, -l, -r$	pizza/N ::$\sim x_5$
money/N ::$\sim x_4$	pizza/N ::$\sim x_5, -l$
money/N ::$\sim x_4, -l$	to/P :: $=x_4, \sim x_5$
pizza/N ::$\sim x_4$	was/T :: $=x_0, +l, \sim x_2$
pizza/N ::$\sim x_4, -l$	what/N ::$\sim x_5, -r$
to/P :: $=x_2, \sim x_2$	what/D ::$\sim x_5, -l, -r$
was/T :: $=x_4, +l, \sim x_4$	who/D ::$\sim x_4, -r$
was/T :: $=x_4, +l, \sim x_0$	who/N ::$\sim x_5, -l, -r$
what/N ::$\sim x_4, -r$	ϵ/v :: $=x_1, \sim x_0$
what/N ::$\sim x_4, -l, -r$	$\epsilon/C_{declarative}$ :: $=x_2, C$
who/D ::$\sim x_2, -r$	$\epsilon/C_{question}$ :: $<=x_2, C$
who/D ::$\sim x_4, -l, -r$	$\epsilon/C_{question}$ :: $<=x_2, +r, C$
ϵ/v :: $=x_4, \sim x_4$	ϵ/v :: $<=x_1, =x_5, \sim x_0$
$\epsilon/C_{question}$ :: $<=x_4, C$	
ϵ/v :: $<=x_4, =x_4, \sim x_4$	
$\epsilon/C_{question}$ :: $<=x_0, +r, C$	
$\epsilon/C_{declarative}$ :: $=x_4, +r, C$	

Table 2: Examples of inferred lexicons that satisfy the conditions imposed by the input sequence in Table-1. Each lexical item has the form, (PF/CAT ::SFS), consisting of a phonetic form (PF), a category (CAT) and a sequence of syntactic features (SFS). The phonetic forms ϵ is covert (unpronounced). The selectional features are $\{x_0, x_1, ..., x_5\}$ and the licensing features are $\{l, r\}$.

(2) translates the annotations for the sentence into (logic) formulas that constrain the parse tree – e.g. predicate-argument relations and morphological agreement are translated into locality constraints[3]; (3) adds these new formulas to the existing system of formulas in S_i to produce S_{i+1}. In order to compute the set of lexicons, $G_i = R(S_i)$, we used the Z3 SMT-solver to solve for the set of lexicons satisfying the formulae in S_i.

Data. The input to the inference procedure is a sequence of fourteen sentences, $I_1 - I_{14}$ in Table-1, each annotated with predicate-argument relations as well as morphological agreement; the sentences listed include passive constructions (I_2, I_4, I_{10}), ditransitive constructions ($I_{11} - I_{14}$), yes/no-questions (I_4, I_6, I_7, I_{10},) and wh-questions (I_1, I_3, I_5, I_8, I_9, I_{11}, I_{13}, I_{14}).

Analysis. We used our procedure to infer a set of minimalist lexicons, denoted here as G^*, from the input sequence described in Table-1. Lexicons sampled from G^* produced parses that do not align with those prescribed by contemporary theories of minimalist syntax. (See Lexicon-A in Table-2 for an example of such a lexicon.)

We filtered out such lexicons by using Z3 to identify lexicons in G^* that were *optimal* with respect to three cost functions that (respectively): (i) *minimized* the number of lexical entries in the lexicon; (ii) *minimized* the total number of selectional and licensing features in the lexicon and the parses (this rewards reduction in the total size of both the lexicon and the parses); (iii) *maximized* the

[3]The *principle of syntactic locality* asserts that syntactic relations are established locally by merge (Sportiche et al., 2013).

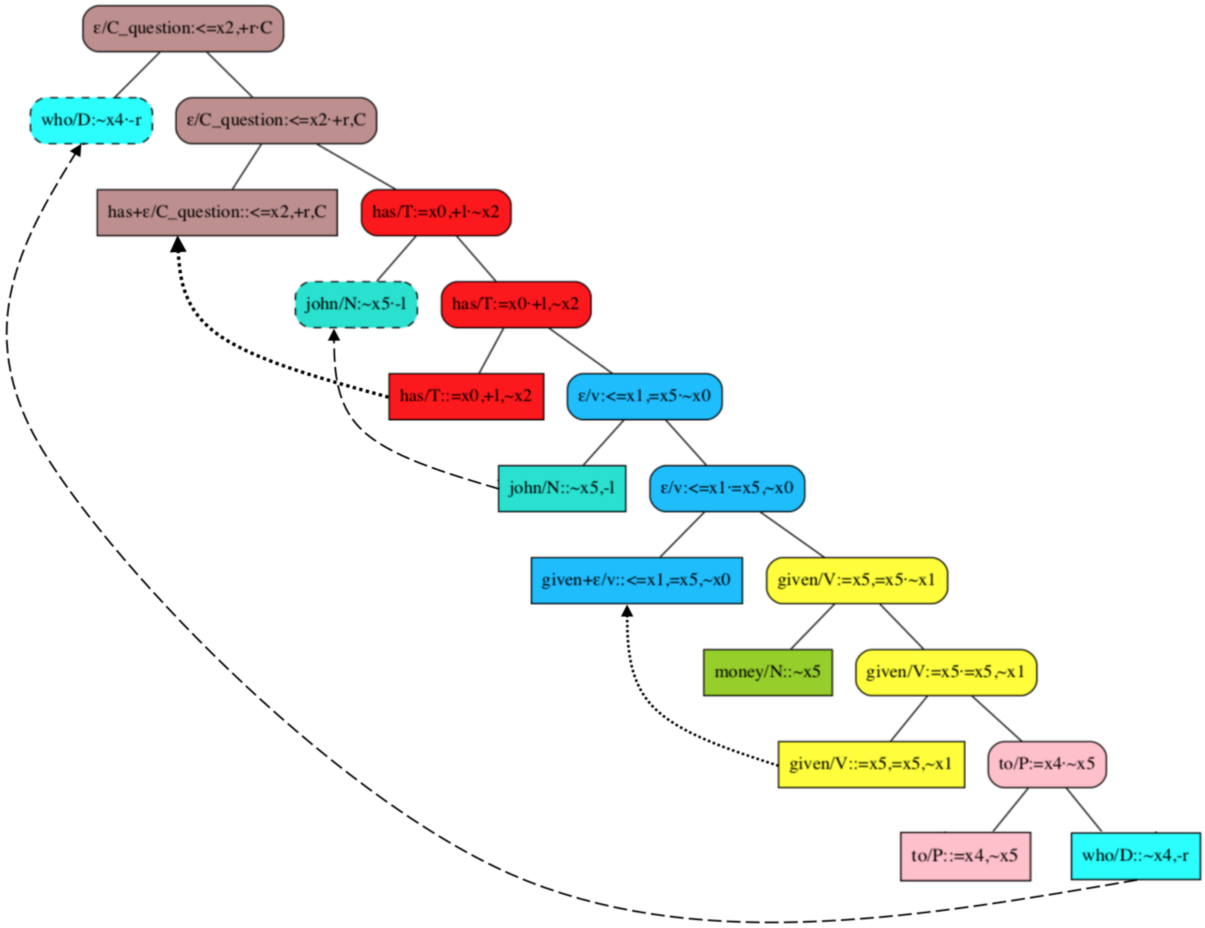


Figure 1: An MG parse for the sentence "Who has John given money to?" (see I_{14} in Table-1 for annotations) derived from Lexicon-B in Table-2. This parse accords with the parse prescribed by contemporary theories of syntax. The feature sequences displayed in non-leaf nodes have a dot, $\cdot$, separating features that have already been consumed (on the left) from those that have not (on the right). The dashed arrows denote phrasal movement. The dotted arrows denote head movement. Nodes with the same *head* have the color. The parse is assembled in a bottom-up manner via merge: "eating" merges with "to who" (formed by first merging "to" and "who") and then with "money", thus establishing (via locality) predicate-argument relations; the resulting structure merges with an empty lexical node with category v, undergoing V-to-v head-movement before merging with the argument "john" in accordance with the VP-Internal Subject Hypothesis (Hale and Keyser, 2002); the resulting structure then merges with the auxiliary verb "has", after which the argument "john" undergoes subject-raising from the VP-shell by (internally) merging with "has", thus establishing morphological agreement between "john" and "has"; next, the head of "has" undergoes T-to-C head-movement to merge with the covert complementizer, $\epsilon/C_{question}$, which indicates that the sentence is an interrogative; finally, "who" undergoes wh-fronting by (internally) merging with $\epsilon/C_{question}$. Wh-fronting (of "who") and Subject-raising (of "john"), instances of A'-movement and A-movement respectively, are triggered by different licensor features, the former by $+r$ and the latter by $+l$.

number of *distinct* selectional features in the lexicon (this rewards lexicons that are more exclusive in which structures they generate).[4] We encoded these cost function as first order logical formulae, adding them to the SMT-solver after running the inference procedure, and then re-solving; the resulting set of (inferred) MGs are optimal with respect to the specified cost functions.

[4]Cost functions (i) and (ii) are based on the MDL principle (see also (Stabler, 1998)), whereas cost function (iii) is based on the Subset principle.

This produced a subset of G^*, denoted F^*, in which each lexicon had exactly: 24 lexical items; 48 features in the lexicon (not including the special feature C); 202 features in the parses; at least five distinct selectional features. Lexicons sampled from F^* produced parses that respect the syntactic relations prescribed in Table-1 and do align with structures prescribed by contemporary theories of minimalist syntax. See Lexicon-B in Table-2 for a representative member of F^* – the syntactic phenomenon that Lexicon-B correctly models includes: A' movement (Wh-fronting for question formation); a (double) VP shell structure that employs V-to-v head-movement (as part of the predicate-argument structure within the parse tree; see (Hale and Keyser, 2002)); T-to-C head-movement (i.e. subj-auxiliary verb inversion) and A-movement (subject raising for morphological agreement). See Figure-1 for a parse produced by Lexicon-B that demonstrates these syntactic phenomenon.

Conclusion. Our results demonstrate that our procedure for inferring MGs is able to acquire knowledge of syntax from psychologically plausible input and employ movement (i.e. displacement) to establish multiple (crossing and nested) discontinuous relations within a syntactic structure. We observe that by enabling and disabling axioms in our model, it is possible to determine which axioms are redundant, and *thereby gain insight into whether the universal linguistic principles, from which the axioms of the system are largely derived, are justified or can be discarded*, thus aiding in the evaluation of the Strong Minimalist Thesis (Chomsky, 2001, 2008). Going forward, we will focus on examining the over-generations produced by the MGs inferred by our procedure and understanding how these over-generations relate to the cost functions used by our procedure for identifying optimal grammars.

Acknowledgements. The author would like to thank Robert C. Berwick, Sandiway Fong, Beracah Yankama, and Norbert Hornstein for their suggestions, feedback, and inspiration.

References

David Adger. 2003. *Core syntax: A minimalist approach*, volume 33. Oxford University Press Oxford.

Clark Barrett and Cesare Tinelli. 2018. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. Springer.

Andrew Barron, Jorma Rissanen, and Bin Yu. 1998. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760.

Robert C. Berwick. 1985. *The acquisition of syntactic knowledge*. MIT press.

Cristian Cadar and Koushik Sen. 2013. Symbolic execution for software testing: three decades later. *Commun. ACM*, 56(2):82–90.

Noam Chomsky. 1965. *Aspects of the theory of syntax*. MIT Press.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press.

Noam Chomsky. 2001. Derivation by phase. In Michael Kenstowicz, editor, *Ken Hale: A life in language*, pages 1–52. MIT press.

Noam Chomsky. 2008. On phases. *Current Studies in Linguistics Series*, 45:133.

Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. TACAS'08/ETAPS'08, pages 337–340. Springer-Verlag.

Leonardo De Moura and Nikolaj Bjørner. 2011. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77.

Thomas Graf. 2013. *Local and transderivational constraints in syntax and semantics*. Ph.D. thesis, University of California at Los Angeles.

Peter D Grünwald. 2007. *The minimum description length principle*. MIT press.

Kenneth L. Hale and Samuel J. Keyser. 2002. *Prolegomenon to a theory of argument structure*, volume 39. MIT press.

Norbert Hornstein, Jairo Nunes, and Kleanthes K Grohmann. 2005. *Understanding minimalism*. Cambridge University Press.

Fernando C. N. Pereira and David H. D. Warren. 1983. Parsing as deduction. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, pages 137–144. Association for Computational Linguistics.

Andrew Radford. 1997. *Syntactic theory and the structure of English: A minimalist approach*. Cambridge University Press.

Manny Rayner, Åsa Hugosson, and Göran Hagert. 1988. Using a logic grammar to learn a lexicon. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 524–529. Association for Computational Linguistics.

James Rogers. 1998. *A descriptive approach to language-theoretic complexity*. CSLI Publications.

Dominique Sportiche, Hilda Koopman, and Edward Stabler. 2013. *An introduction to syntactic analysis and theory*. John Wiley & Sons.

Edward Stabler. 1996. Derivational minimalism. In *Intl. Conf. on Logical Aspects of Comp. Ling.*, pages 68–95. Springer.

Edward P Stabler. 1993. *The Logical Approach to Syntax*. MIT Press.

Edward P. Stabler. 1998. Acquiring languages with movement. *Syntax*, 1(1):72–97.

Edward P. Stabler and Edward L Keenan. 2003. Structural similarity within and among languages. *Theoretical Computer Science*, 293(2):345–363.

Kenneth Wexler. 1993. The subset principle is an intensional principle. In *Knowledge and language*, pages 217–239. Springer.

Extending adaptor grammars to learn phonological alternations

Canaan Breiss
University of California, Los Angeles
cbreiss@ucla.edu

Colin Wilson
Johns Hopkins University
colin@cogsci.jhu.edu

1 Overview

Recent advances in unsupervised learning of linguistic structure have demonstrated the feasibility of inferring latent morphological parses from an unannotated corpus given transparent underlying-to-surface mappings (ex., Adaptor Grammars (AGs); (Johnson et al., 2007; Johnson and Goldwater, 2009), as well as in learning predictable phonological transformations from prespecified underlying morphemes to a range of surface allomorphs via a stochastic edit distance algorithm (Cotterell et al., 2015). In this paper we introduce a nonparametric Bayesian model which builds on the morpheme-segmentation success of AGs, and incorporates the ability to learn predictable phonological transformations of underlying forms to their surface allomorphs via the interaction of markedness and faithfulness principles, inspired by generative phonology. The unsupervised nature of this model (that is, no semantic information about the words being segmented is provided) is relevant not only computationally but also psychologically, as it mirrors developmental findings (Kim, 2015) that young infants segment and cluster morphemes based solely on phonetic and distributional similarity. The model also incorporates many of the other cognitive restrictions infants during the initial period of morphophonological learning in an effort to make the model maximally realistic, and thus eventually useful in making quantitative predictions about the early stages of morphophonological acquisition that can be experimentally investigated. We evaluate the model on a novel dataset consisting of a complex system of allomorphy in Acehnese, an understudied Indonesian language.

2 Model design

The model takes the general structure of a (relatively shallow) AG with rewrite rules Word $\rightarrow$ Morph(s), Morph $\rightarrow$ Phoneme(s). The model differs, however, in that it considers whether a possible novel morpheme could be derived from an existing item in the lexicon via a phonological transformation (at a cost), as well as reused directly (if it exactly matches a lexical item) or generated anew. The parameterization of the penalty for non-identity transformations is informed by research demonstrating that infant and adult learners prefer phonetically-minimal alternations (ex., White (2013), cf. (Steriade, 2009) on the P-Map hypothesis more broadly), and that speakers are sensitive to the segment-to-segment transitional probabilities (cf. Vitevitch and Luce (2004)) of their native language(s). Thus, the probability of a novel morpheme being a transformation of an existing one is equal to the probability of the source morpheme in question being reused (as in a standard AG) multiplied by the penalty associated with a specific segment-to-segment mapping, operationalized as the number of phonological feature values by which the input and output segments differ ("faithfulness" to the input). This quantity is then multiplied by the probability of the surface string created through the unfaithful mapping, as calculated from the surface-distribution of phonemes in the unsegmented corpus (corresponding to a penalty for the "markedness" of the surface form), and the morpheme-length parameter λ. The faithfulness penalties on segment-to-segment transformations was equal to twice the featural edit distance between the two segments, and penalties for surface forms were calculated via segmental trigram probabilities of the corpus.

2.1 Implementation

Unless otherwise noted below, the model was initialized with words parsed as monomorphemic roots, following the phonological acquisition literature which shows infants store unanalyzed chunks of their input during early learning (Ngon et al., 2013). Inference for all parameters was carried out via Gibbs sampling; the hyperparameters α and β, as well as the length penalty λ on morpheme lengths, were sampled using the slice-sampling technique from Neal et al. (2003), as implemented in Johnson and Goldwater (2009).

3 Data

We tested the model on a group of morphophonological alternations observed affecting labial-initial prefixes in Acehnese (Malayo-Polynesian, 3.5 million speakers, primarily in Indonesia). Two Acehnese verbal prefixes *peu-* /pɯ-/ and *meu-* /mɯ-/ exhibit allomorphy when prefixed to a base which begins with a labial consonant ({/p, b, m, w/}), surfacing as to [pu-] and [mu-] respectively with the back high unrounded vowel having undergone the phonological process *rounding*. A second process, *spirantization*, applies to the *peu-* prefix when the base to which it is attached begins with a labial consonant and is also polysyllabic, changing the initial consonant of the prefix from /p/ to [s], as in /pɯ-maŋat/ → [sɯmaŋat]. Further, spirantization *bleeds* rounding when the conditioning environments overlap, appearing to "apply" beforehand and so removing the environment (the labial onset of the prefix) which would have triggered rounding: /pɯ-maŋat/ → [sɯmaŋat], *[sumaŋat] (Durie, 1985). Thus, summarizing the data pattern, we find: /pɯ-/ → {[pɯ-, pu-, sɯ-]}, /mɯ-/ → {[mɯ-, mu-]}.

The use of Acehnese in evaluating the model is relevant for two reasons. First, there has been no known computational work on the language, nor even detailed quantitative study of the languages morphophonology. Therefore, the phenomena explored here (idealized based on corpus data gathered as part of Breiss, in prep.) provide a novel perspective on which to test traditionally English-centric tests of unsupervised learning of linguistic structure. Secondly, the specifics of the morphophonological alternations in the Acehnese data are typologically unusual, exhibiting processes which are both phonetically-motivated (rounding in the context of two labial segments) as well as phonetically arbitrary (spirantization). Prior research has shown that speakers may be biased towards learning and/or generalizing phonetically-natural patterns or processes more than phonetically-arbitrary ones; therefore, the trade-off in productivity between lexical listing and phonological derivation of allomorphs instantiated in the model can be used to make testable, quantitative predictions about human behavior.

	None	Half	All
Segmentation only			
Morpheme	1 / 0.45 / 0.62	1/ 0.87 / 0.93	1 / 1 / 1
Boundary	0.76 / 0.09 / 0.17	1 / 0.47 / 0.64	1 / 1 / 1
Source	(n/a)	(n/a)	(n/a)
Allomorphy only			
Morpheme	(n/a)	(n/a)	(n/a)
Boundary	(n/a)	(n/a)	(n/a)
Source	100% / 100 %	100% / 100 %	100% / 100 %
Both			
Morpheme	1 / 0.45 / 0.62	1 / 0.88 / 0.93	1 / 0.99 / 0.99
Boundary	0.76 / 0.09 / 0.17	1 / 0.51 / 0.67	1 / 0.98 / 0.99
Source	100% / 85%	100% / 70%	100% / 100 %

Figure 1: Evaluation statistics; each cell displays Precision / Recall / F-score for that combination of model settings and data.

4 Evaluation

F-score for identifying polymorphemic words, morpheme boundary F-score, and the percentage of surface allomorphs were derived from the correct underlying form (prefix and root) were calculated. We test each of the methods on a dataset consisting solely of polymorphemic words, a dataset with bare roots for 50% of the polymorphemic words, and a dataset with bare roots for all of the polymorphemic words (referred to as Zero, Half, and All respectively). Results are presented in 1, where each cell lists Precision / Recall / F-score.

4.1 Segmentation only

The first test is whether, under ideal conditions, the model correctly parses the data into its surface allomorphs. Disabling the option to consider non-faithful lexical reuse, the model is able to perform moderately well on segmenting the corpus. Since the Zero setting did not discover any segmentation with words initialized as unanalyzed roots, random initialization was used for this condition only.

4.2 Allomorphy only

The phonological corollary to the morphological segmentation question is whether, under ideal conditions, the model can collapse the allomorphs of each morpheme into a single underlying representation. For this test, we gave the morphemic parse of each of the words in training, and then allowed the model to be informed by the faithfulness penalties as it discovered the most likely division between lexicalization and derivation for each of the allomorphs.

4.3 Simultaneous morphological segmentation and phonological abstraction over allomorphs

We test the model in a more realistic situation by asking it to discover the correct segmentation as well as the correct phonological alternations, and find that neither task is impaired when performed jointly with the other (in fact, in certain cases the performance is marginally improved; we take this as a suggestion that further scaling up of the model and dataset may give rise to more robust synergies; cf. Johnson (2008)).

5 Future work

While the model as presented here represents a significant step towards integrating insights from the developmental literature with computational methods of learning of linguistic structure from unlabelled data, it is hardly an adequate or complete model of early morphophonological acquisition. We see three main fronts along which the model can be improved: robustness to (more) naturalistic data, greater flexibility in non-faithful transformations to handle epenthesis and deletion phenomena, and the more robust integration of phonological principles to yield interpretable constraint-based grammars as part of the model yield.

In terms of data realism, the model can be improved so as to handle noisier, larger datasets: while the model does well given at least *some* bound-free pairs as evidence, not all languages allow roots to surface bare; thus improving the willingness of the model to consider morphological decomposition even in the absence of minimal pairs of bound-affixed forms is essential.

The linguistic validity of the range of hypotheses that the model considers can be enhanced by allowing it to consider strings of varying lengths as possible sources for non-faithful transformations. As implemented, the model only considers non-identity lexical sources for novel morphemes which are of the same length (in phonemes) as the novel morpheme under consideration. However, natural languages frequently exhibit deletion or epenthesis processes as part of morphophonological alternations (ex., the allomorphy of the English plural; /-z/ $\rightarrow$ {[-z, -s, -əz]}).

Two further improvements to the way that the model handles markedness and faithfulness penalties will allow the trained model to yield a grammar of weighted constraints, in addition to a lexicon and morphological parse, which can be compared to those which are the subject of analysis in other areas of generative phonology. On the faithfulness front, future experimental work can ground the specific penalties associated with non-identity transformations in data from confusability matrices, as in White (2017). These findings can be incorporated into the model by treating the phonetic distance between non-faithful mappings of segments as the mean of a Gaussian prior over possible penalties, rather than an absolute penalty itself. This will allow the model to deviate from the phonetically-informed priors in the face of compelling language-specific evidence for phonetically-unnatural alternations, mirroring the experimental findings of Wilson (2006).

The phonotactic markedness penalty given to surface forms can be enhanced by incorporating the ability to learn language-specific, feature-based phonotactic constraints from the already-segmented lexicon. This is motivated by the work ofHayes and Wilson (2008); Becker et al. (2011); Kager and Pater (2012); Hayes and White (2013); Rasin and Katzir (2016), among others, which shows that adult speakers internalize only a subset of available statistical generalizations latent in the data, informed by the statistics of the language and possibly prior grammatical knowledge. This constraint-based markedness penalty would replace the current phoneme trigram penalty over surface forms.

References

Michael Becker, Nihan Ketrez, and Andrew Nevins. 2011. The surfeit of the stimulus: Analytic biases filter lexical statistics in turkish laryngeal alternations. *Language*, 87(1):84–125.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.

Mark Durie. 1985. *A grammar of Acehnese on the basis of a dialect of North Aceh*, volume 111. Mark Durie.

Bruce Hayes and James White. 2013. Phonological naturalness and phonotactic learning. *Linguistic Inquiry*, 44(1):45–75.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Mark Johnson. 2008. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of ACL-08: HLT*, pages 398–406.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325. Association for Computational Linguistics.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in neural information processing systems*, pages 641–648.

René Kager and Joe Pater. 2012. Phonotactics as phonology: Knowledge of a complex restriction in dutch. *Phonology*, 29(1):81–111.

Yun Jung Kim. 2015. *6-month-olds' segmentation and representation of morphologically complex words*. Ph.D. thesis, UCLA.

Radford M Neal et al. 2003. Slice sampling. *The annals of statistics*, 31(3):705–767.

Céline Ngon, Andrew Martin, Emmanuel Dupoux, Dominique Cabrol, Michel Dutat, and Sharon Peperkamp. 2013. (non) words,(non) words,(non) words: evidence for a protolexicon during the first year of life. *Developmental Science*, 16(1):24–34.

Ezer Rasin and Roni Katzir. 2016. On evaluation metrics in optimality theory. *Linguistic Inquiry*, 47(2):235–282.

Donca Steriade. 2009. The phonology of perceptibility effects: The p-map and its consequences for constraint organization. the nature of the word: essays in honor of paul kiparsky, ed. by kristin hanson and sharon inkelas.

Michael S Vitevitch and Paul A Luce. 2004. A web-based interface to calculate phonotactic probability for words and nonwords in english. *Behavior Research Methods, Instruments, & Computers*, 36(3):481–487.

James White. 2017. Accounting for the learnability of saltation in phonological theory: A maximum entropy model with a p-map bias. *Language*, 93(1):1–36.

James Clifford White. 2013. *Bias in phonological learning: Evidence from saltation*. Ph.D. thesis, UCLA.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive science*, 30(5):945–982.

Induction of Minimalist Grammars over Morphemes

Marina Ermolaeva
University of Chicago
`mermolaeva@uchicago.edu`

1 Introduction

Syntactic literature tends towards a big-picture outlook, abstracting away from details such as full specifications of lexical items or features involved in derivations. However, a lower-level description is required to identify differences between competing analyses of the same phenomenon.

For a concrete example, consider the double object construction (e.g. *John gave Mary a book*) in English. One option is to combine the internal arguments *Mary* and *a book* in a "small clause" or PP-like structure and then merge the verb with this constituent (e.g. Kayne 1984; Pesetsky 1996; Harley and Jung 2015). The alternative is to have the verb select the arguments one by one, giving rise to VP-shells (Larson, 1988) and analyses inspired by them (Kawakami, 2018).

It is natural to ask whether it would be possible, assuming a sufficiently rich formalism compatible with the Minimalist framework, to choose the answer to this and similar questions based on some robust quantitative metric.

2 Minimalist grammars

Minimalist grammars (Stabler, 1997) are a natural choice for this task. As a formalization of Chomsky's (1995) Minimalist Program, they are well-suited for implementing analyses of syntactic phenomena, yet at the same time explicit regarding the assumptions about syntactic units and operations.

Minimalist grammars define lexical items (atomic expressions) as pairs consisting of a phonetic exponent and a sequence of syntactic features (1). The first feature of each lexical item is accessible to the operations, Merge and Move, that target and delete matching features of opposing polarities. Merge combines two expressions to build a new one, whereas Move is unary and attracts a sub-expression into the specifier of the main structure. Merge with head movement (HM) concatenates pronounced features of the heads of its arguments, providing a simple implementation of concatenative morphology.

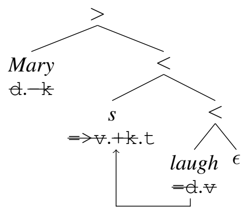
(1)

	Positive polarity	Negative polarity
Merge	`=x` (right selector) `=>x` (HM selector) `x=` (left selector)	`x` (category)
Move	`+x` (licensor)	`−x` (licensee)

Whichever expression contributed the positive feature becomes the head of the new expression. A complete sentence is an expression with no features left but the category `t` on its head. An example lexicon is given in (2), along with the derived tree of the sentence *Mary laughs* generated by it.

(2)

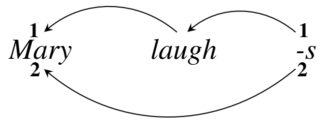
Mary :: `d.−k`
s :: `=>v.+k.t`
ed :: `=>v.+k.t`
laugh :: `=d.v`
jump :: `=d.v`



3 Learning from dependencies

There is a substantial body of work dedicated to learning grammars from unstructured strings; e.g. an overview in (Clark, 2017). In particular, Yoshinaka (2011) presents an algorithm for learning certain subclasses of multiple context-free grammars. One can construct an equivalent Minimalist grammar for any multiple context-free grammar (Michaelis, 2001). However, such a grammar would not make for a good starting point if our goal is to compare and evaluate proposals of theoretical syntax, as modern syntactic theory heavily relies on highly abstract concepts such as empty categories, not directly visible in the raw data.

On the other hand, Siskind (1996) suggests that rather than obtain syntactic structure from unstructured input, the learner can start the process of *grounding*, or mapping linguistic units to atoms of meaning, before learning syntax. Then it is plausible that the learner can identify relations formed by Merge and Move before knowing what lexical items or syntactic features are involved, which gives rise to the approach to learning proposed by Kobele et al. (2002). For each sentence the learner is given ordered and directed dependencies between morphemes, with suffixes marked as such (3).

(3) 

In this scenario, full lexical items (unique for each sentence) can be recovered from the dependencies. The learner's task is to determine which feature distinctions should be kept and which need to be collapsed, or unified. The pressure for unification comes from a restriction on the number of homophonous lexical items (Kanazawa, 1995).

As an illustration, consider the corpus of two sentences, *Mary laugh -s* and *Mary laugh -ed*. The learner assembles lexical items by assigning a fresh feature to each dependency, assuming that each data point is a complete sentence of category `t`. The ordering of dependencies determines whether each of them corresponds to Merge or Move. The initial lexicon (4) contains two copies each of *Mary* and *laugh*.

(4)

Mary :: `f1.-f2`	*Mary* :: `f4.-f5`
laugh :: `=f1.f3`	*laugh* :: `=f4.f6`
-s :: `=>f3.+f2.t`	*-ed* :: `=>f6.+f5.t`

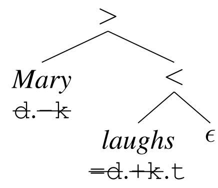
The final step is to rename the corresponding features throughout the lexicon in order to collapse each pair of items into one. A familiar-looking lexicon will arise if `f1` and `f4` are mapped to `d`, `f2` and `f5` to `k`, and `f3` and `f6` to `v`. After feature unification, the grammar shrinks from six to four lexical items, which can still derive the input sentences.

4 Lexical item decomposition

This paper builds on (Kobele et al., 2002), aiming to relax the segmentation requirement and let the algorithm learn the structure within complex words and any generalizations it would lead to.

Compare the lexicon in (2) with (5), which generates exactly the same set of sentences. Intuitively, (2) is better than (5), even though both have the same number of lexical items. It captures the similarities between different forms of the same verb and recognizes the verbs' internal structure: two correct generalizations that (5) misses.

(5)
Mary :: `d.-k`
laughs :: `=d.+k.t`
laughed :: `=d.+k.t`
jumps :: `=d.+k.t`
jumped :: `=d.+k.t`



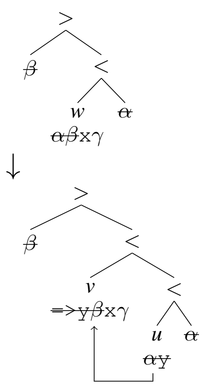
This difference can be quantified in a number of ways – naively as the number of phonetic and/or syntactic units, length of encoding the grammar or, taking into account the cost of encoding the corpus, as minimum description length (Rissanen, 1978).

How to transition from a grammar over words such as (5) to a grammar over morphemes (2)? In linguistic terms, the latter reanalyzes the verb as a complex head formed by head movement. This can be generalized to a *decomposition* operation (Kobele, 2018) that splits a lexical item's syntactic and phonetic features, producing a new item with a fresh category (6). The morphological operation generating w from the stem u and suffix v is denoted by $\oplus$; in the simplest case it corresponds to string concatenation.

(6)
w :: $\alpha\beta$xγ

$\downarrow$

u :: αy
v :: =>yβxγ
$w = u \oplus v$



If syntactic decomposition is not accompanied by splitting the phonological material, one of the new lexical items will be an empty functional head. Otherwise, the algorithm has to construct a morphological rule by searching for phonological similarities across the lexicon.

Concatenative morphology has been shown to be successfully learnable in an unsupervised scenario (Goldsmith, 2001), with a possibility of using the results to infer the syntactic category of words (Hu et al., 2005); the problem of irregular and non-concatenative patterns (such as *sings* vs. *sang*) is also addressed in the literature (e.g. Lee and Goldsmith 2014). Thus, in our case the learner has access to two separate sources of information – syntactic features and phonological patterns – to base its decisions on.

Multiple lexical items sharing a sub-sequence of syntactic features can be decomposed simultaneously, factoring out the shared features. The pressure to do this comes from a reduced cost in features; replacing repeating sequences is a well-known compression technique (cf. Nevill-Manning et al. 1994).

5 Towards a grammar over morphemes

The following example shows how a naive word-based grammar can be transformed into a linguistically motivated grammar over morphemes via decomposition and feature unification. Let the learner start with dependency structures (over non-segmented words) for the following eight sentences:

(7)

Mary laughs	*Mary jumps*
Mary is laughing	*Mary is jumping*
Mary will laugh	*Mary will jump*
Mary will be laughing	*Mary will be jumping*

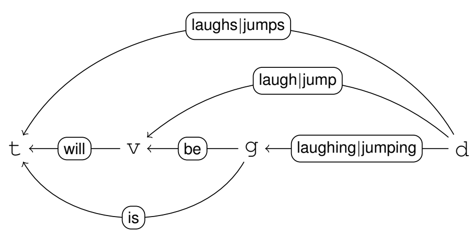
From this data set, the algorithm discussed in section 3 can extract the lexical items shown in (8) by collapsing homophonous items.

(8)

	laughs :: =d.+k.t
Mary :: d.-k	*laughing* :: =d.g
is :: =g.+k.t	*laugh* :: =d.v
will :: =v.+k.t	*jumps* :: =d.+k.t
be :: =g.v	*jumping* :: =d.g
	jump :: =d.v

Merge dependencies in this lexicon can be conveniently visualized as a directed graph. In (9) vertices are category features; each edge corresponds to a lexical item and connects the category of its complement (first phrase it selects) to that of its own.

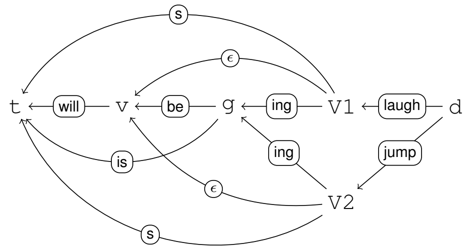
(9)



We begin by decomposing lexical verbs, producing the lexicon in (10). The three lexical items *laughs*, *laughing*, and *laugh* are a valid target for decomposition; and so are *jumps*, *jumping*, and *jump*. Both transitions are motivated both phonologically (factoring out a common prefix) and syntactically (splitting three feature bundles starting with =d).

(10)

Mary :: d.-k	*s* :: =>V1.+k.t
is :: =g.+k.t	*s* :: =>V2.+k.t
will :: =v.+k.t	*ing* :: =>V1.g
be :: =g.v	*ing* :: =>V2.g
laugh :: =d.V1	ϵ :: =>V1.v
jump :: =d.V2	ϵ :: =>V2.v

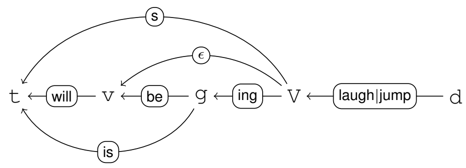


$laughing = laugh \oplus ing$	$laughs = laugh \oplus s$
$jumping = jump \oplus ing$	$jumps = jump \oplus s$

This move created two copies each of *s*, *ing*, and ϵ. All of them can be conflated by unifying a single pair of features, V1 and V2, producing a much smaller grammar (11).

(11)

	laugh :: =d.V
Mary :: d.-k	*jump* :: =d.V
is :: =g.+k.t	*s* :: =V.+k.t
will :: =v.+k.t	*ing* :: =V.g
be :: =g.v	ϵ :: =V.v

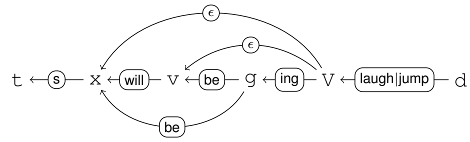


$laughing = laugh \oplus ing$	$laughs = laugh \oplus s$
$jumping = jump \oplus ing$	$jumps = jump \oplus s$

The next step targets another repeated sequence of syntactic features: +d.t. This essentially creates a dedicated Tense projection, which hosts the surface position of the subject (12). At this point, concatenation is no longer sufficient for the morphological rules, highlighting the need for a richer theory of morphology.

(12)

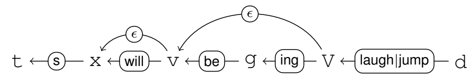
Mary :: d.-k	*laugh* :: =d.V
s :: =x.+k.t	*jump* :: =d.V
be :: =g.x	ϵ :: =V.x
will :: =v.x	*ing* :: =V.g
be :: =g.v	ϵ :: =V.v



$$\begin{array}{rr} & laughs = laugh \oplus s \\ laughing = laugh \oplus ing & jumps = jump \oplus s \\ jumping = jump \oplus ing & is = be \oplus s \\ & will = will \oplus s \end{array}$$

This grammar still contains two copies of *be*. While they could be collapsed by unifying `v` and `x`, this move would cause the grammar to overgenerate, producing, for example, the set of ungrammatical sentences *Mary* (*will*)$^+$ *be laughing*. However, adding an edge (empty head) from `v` to `x` would make two of these items redundant without generating any unwanted sentences (13). This move can be thought of as decomposing *be* `::` `=g.x` into *be* `::` `=>g.z` and ϵ `::` `=z.x`, where `z` is a fresh feature, and then unifying `z` with `v`. The same is applicable to ϵ `::` `=V.x` and ϵ `::` `=V.v`.

(13)
$$\begin{array}{rlrl} Mary & :: \texttt{d.-k} & & \\ s & :: \texttt{=x.+k.t} & laugh & :: \texttt{=d.V} \\ \epsilon & :: \texttt{=v.x} & jump & :: \texttt{=d.V} \\ will & :: \texttt{=v.x} & ing & :: \texttt{=V.g} \\ be & :: \texttt{=g.v} & \epsilon & :: \texttt{=V.v} \end{array}$$



$$\begin{array}{rr} & laughs = laugh \oplus s \\ laughing = laugh \oplus ing & jumps = jump \oplus s \\ jumping = jump \oplus ing & is = be \oplus s \\ & will = will \oplus s \end{array}$$

We have shown how a Minimalist grammar can be compressed in a way compatible with linguistic theory through repeated application of lexical item decomposition and feature unification. Together they offer a principled way to identify repeating patterns in the lexicon, instantiate them as new lexical items, and collapse any emerging duplicates. Our current work in progress involves building a learning algorithm for syntax with these two operations at its core. This approach would allow to derive (potentially empty) functional heads, producing linguistically motivated generalizations.

References

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press.

Alexander Clark. 2017. Computational learning of syntax. *Annual Review of Linguistics*, 3:107–123.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

Heidi Harley and Hyun Kyoung Jung. 2015. In support of the $\mathrm{p_{HAVE}}$ analysis of the double object construction. *Linguistic inquiry*, 46(4):703–730.

Yu Hu, Irina Matveeva, John Goldsmith, and Colin Sprague. 2005. Using morphology and syntax together in unsupervised learning. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 20–27. Association for Computational Linguistics.

Makoto Kanazawa. 1995. *Learnable Classes of Categorial Grammars*. Ph.D. thesis, Stanford University.

Masahiro Kawakami. 2018. Double object constructions: Against the small clause analysis. *Journal of Humanities and Social Sciences*, 45:209–226.

Richard S. Kayne. 1984. *Connectedness and Binary Branching*. Foris, Dordrecht.

Gregory M. Kobele. 2018. Lexical decomposition. *Computational Syntax* lecture notes.

Gregory M. Kobele, Travis Collier, Charles Taylor, and Edward P. Stabler. 2002. Learning mirror theory. In *Proceedings of TAG+ 6*, pages 66–73.

Richard K Larson. 1988. On the double object construction. *Linguistic inquiry*, 19(3):335–391.

Jackson Lee and John Goldsmith. 2014. Automatic morphological alignment and clustering. Technical report, Technical report TR-2014-07, Department of Computer Science, University of Chicago.

Jens Michaelis. 2001. *On formal properties of minimalist grammars*. Ph.D. thesis, U of Potsdam.

Craig G Nevill-Manning, Ian H Witten, and David L Maulsby. 1994. Compression by induction of hierarchical grammars. In *Proceedings of DCC'94*, pages 244–253.

David Michael Pesetsky. 1996. *Zero syntax: Experiencers and cascades*. MIT press.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Jeffrey Mark Siskind. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1-2):39–91.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *Selected Papers from LACL '96*, pages 68–95. Springer Berlin Heidelberg.

Ryo Yoshinaka. 2011. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science*, 412(19):1821–1831.

Immature representation or immature deployment? Modeling child pronoun resolution

Hannah Forsythe
University of California, Irvine
SSPB 2211, Irvine, CA 92617
hforsyth@uci.edu

Lisa Pearl
University of California, Irvine
SSPB 2219, Irvine, CA 92617
lpearl@uci.edu

Abstract

Children acquiring Spanish interpret subject pronouns differently from adults, initially relying on pragmatic cues instead of morphosyntactic cues that are more statistically reliable. Following Gagliardi et al. (2017), we use Bayesian cognitive modeling to explore the sources of this non-adult-like behavior, investigating whether it is more likely due to (i) noise in children's representation of the probability that some cues favor certain antecedents, or (ii) noise in children's deployment of otherwise adult-like probabilities. Results favor noisy deployment as the source of children's non-adult-like pronoun resolution.

1 Intro

When children produce and interpret language differently from adults, the underlying cause can be unclear: do children have an immature representation of the target language, or do they simply deploy that representation in an immature way? One way to get at this question is to design behavioral tasks that facilitate deployment (e.g., lowering processing demands, improving task pragmatics, using more sensitive behavioral measures), with the idea that any non-adult-like behavior that remains after deployment effects have been controlled for is likely due to representational issues. However, it can be difficult to know for sure that deployment effects have been completely controlled for in any given experiment.

Here, we show how cognitive modeling can be used to more directly target representational versus deployment explanations of children's non-adult-like behavior. Following the approach of Gagliardi et al. (2017), we use Bayesian models on a case study of Spanish-speaking children's pronoun resolution to explore whether their non-adult-like use of pronominal cues is better modeled as noise in (i) the representation of the information these cues provide, or (ii) the deployment of that information during interpretation. Results suggest that noisy deployment is more likely to underlie children's non-adult-like behavior in this case. We also discuss implications for both the development of pronoun knowledge, and the investigation of linguistic development more generally.

2 Non-adult-like pronoun resolution by children acquiring Spanish

In Spanish, the interpretation of subject pronouns in context depends on many constraints, but some constraints are stronger than others. For instance, subject pronouns can be probabilistically biased by cues such as an accompanying discourse **CONN**ective or by the speaker's choice of pronominal **FORM**. In (1), temporal connective *y después* ('and then') favors the subject antecedent (*la maestra*: 'the teacher') more strongly than causal *porque* ('because'), and use of the null subject (*pro*) favors the subject antecedent more strongly than the overt pronoun (*ella*). In contrast to these probabilistic cues, subject pronouns can be categorically disambiguated by the accompanying verb's agreement **MOR**phology. In (2), -3S indicates the singular subject while -3P indicates the plural object.

(1) La maestra saluda a la niña, (y después / porque) Ø /ella sale
the teacher waves at the girls, (and then / because) *pro*/she leaves

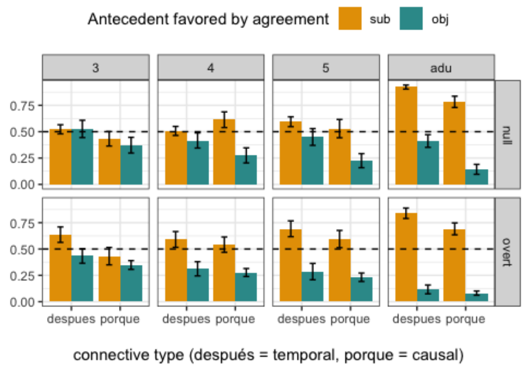
'The teacher waves at the girl, (and then/because) PRONOUN leave(s).'

(2) La maestra$_i$ saluda a las niñas$_k$, y Ø sale$_i$ / salen$_k$
the teacher waves at the girls, and *pro* leave-3S / leave-3P

'The teacher waves at the girls, and PRONOUN leave(s).'

To determine how Spanish-speaking adults and children use these three cues to interpret pronouns in context, Forsythe (accepted) used a forced-choice picture selection task with 47 adults, and 98 preschoolers. Participants listened to sentences like (1) and (2) and indicated their interpretation of the pronoun by choosing an illustration depicting either the subject interpretation (e.g., the teacher waving) or the non-subject interpretation (e.g., girls waving). Cues were fully crossed, systematically aligning and pitting each type against the other two. Figure 1 shows how often participants chose the subject interpretation.

Figure 1: How often children ages 3-5 and adults favor the subject interpretation of a pronoun in context, given different cues: connectives (*después, porque*), pronoun form (*null*, *overt*), and agreement morphology (agreeing with *sub*ject or *obj*ect).



Adults favor the expected interpretation (subject vs. non-subject) on the basis of agreement morphology, but interestingly, this preference is not completely categorical: it is modulated by the cues of connective and pronominal form, which probabilistically bias the interpretation towards or away from the subject antecedent. That is, adults rely on all three cues when interpreting pronouns. Children's behavior shows qualitatively different patterns, with three-year-olds relying only on connectives, four-year-olds relying only on morphology, and five-year-olds relying on both morphology and connectives. Importantly, it is unclear from these results whether children's non-adult-like pronoun interpretation behavior is due to an immature representation of the information that these cues carry (e.g., three-year-olds only have an adult-like representation of connectives) or to an immature ability to deploy the representations they have (e.g., three-year-olds have an adult-like representation of morphology, but fail to access it correctly in the moment). This is where cognitive modeling can help.

3 Modeling child pronoun resolution behavior

This pronoun interpretation behavior serves as our modeling target: a successful model will match children's behavioral patterns in each experimental condition as closely as possible. The model's input will be the same input that children acquiring Spanish use when learning how each of these cues predicts pronoun antecedents. Table 1 shows the rate of reference to the preceding subject antecedent and to singular antecedents, for different cue types, based on samples drawn from a corpus of 54,757 child-directed Spanish utterances.

Table 1: Rates of reference to different antecedent types in the presence of different CONNectives, pronoun FORMs, and agreement MORphology in child-directed Spanish.

cue	value	subject antecedent
CONN	*después*	(29/54) 54%
	porque	(52/149) 35%
FORM	null	(1,093/2,376) 46%
	overt	(64/291) 22%
		singular antecedent
MOR	singular	(5,655/5,662) 99.9%
	plural	(9/1,336) 0.7%

All the cues in these child-directed speech samples appear to follow the patterns we expect from adult behavior: connectives and pronominal form are more probabilistic cues, while agreement morphology is fairly categorical. This input pattern makes it surprising that younger children initially don't rely on agreement morphology.

To probe the underlying source of this immature behavior, we follow Gagliardi et al. (2017), who model linguistic immaturity as noise–either noise in the modeled child's representation of the information a given cue provides, or noise in the ability to reliably use that information in novel situations, such as an experimental task. Here, we ask whether children's non-adult-like interpretation of Spanish pronouns is best captured as noise in the representation of the information provided by cues from connectives, pronominal form, and agreement morphology, or as noise in how this information is accessed during the experiment.

3.1 Baseline model

We model children's reasoning process, which combines the information provided by cues in the child's input with the child's prior about the pronoun's most likely antecedent, using Bayesian inference as in (1). Bayesian inference is often used for cognitive development modeling, as it can capture human behavior very well (e.g., Perfors et al. (2011) Pearl and Mis (2016)).

The modeled child calculates the probability of a potential pronoun antecedent α (e.g., the teacher) given a particular combination of cues extracted from the pronoun and its utterance (e.g., f_{MOR}:sg, f_{CONN}: *después*, f_{FORM}: null), which corresponds to the posterior $P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}})$. This posterior is calculated by considering two probabilities extracted from the input: (i) the likelihood of each cue's value, given that type of antecedent ($P(f_{\text{CUE\_VAL}}|\alpha)$) and (ii) the prior probability of referring to this type of antecedent ($P(\alpha)$).

$$P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}}) \propto P(f_{\text{MOR}}|\alpha) \cdot P(f_{\text{CONN}}|\alpha) \cdot P(f_{\text{FORM}}|\alpha) \cdot P(\alpha) \quad (1)$$

This version of the modeled child makes optimal use of the cues as they appear in the input and will therefore rely most heavily on the most reliable cues, such as morphology–in clear contrast to what we observe in children. To model a child with either immature representations of cue information or immature deployment of cue information, we introduce noise into this optimal model.

3.2 A noisy representation model

The noisy representation model we implement encodes the idea that children behave differently from adults because they have an immature representation of one or more pronominal cues, which is caused by noisily extracting cue information from the input. For example, suppose the link between singular and plural surface agreement and underlying number features is immature. This would prevent the child from accurately tracking how the number semantics of a pronoun's accompanying agreement marker predicts the number semantics of its antecedent (i.e., for the child, singular morphology might not categorically predict a singular antecedent). This in turn could flatten the dramatic difference between $P(\alpha\text{:sg}|f_{\text{MOR}}\text{:sg})$ and $P(\alpha\text{:sg}|f_{\text{MOR}}\text{:pl})$ that is evident from the Spanish-language input in Table 1. Whatever the cause, noisy encoding of cue information from the input will yield non-adult-like likelihood terms.

The noisy representation model in (2) flattens the distributions for each likelihood term using softmax ($e^{\sigma \cdot P}$), which is standardly used for this purpose to model decision-making tasks, including language tasks (e.g. Frank and Goodman (2012); Goodman and Stuhlmüller (2013); Scontras and Goodman (2017)). The level of noise associated with each cue type is controlled by the parameters σ_{MOR}, σ_{CONN}, and σ_{FORM}, with smaller values indicating a flatter distribution and greater values indicating a sharpened distribution.

$$P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}}) \propto e^{\sigma_{\text{MOR}} P(f_{\text{MOR}}|\alpha)} \cdot e^{\sigma_{\text{CONN}} P(f_{\text{CONN}}|\alpha)} \cdot e^{\sigma_{\text{FORM}} P(f_{\text{FORM}}|\alpha)} \cdot P(\alpha) \quad (2)$$

3.3 Two noisy deployment models

We also implement two noisy deployment models encoding the idea that children behave differently from adults because they immaturely access adult-like cue representations during the experimental task. Both models accurately encode the cue information from children's input but deploy this information inaccurately, either (i) occasionally deleting cue information (*noisy deletion*), or (ii) substituting accurate cue information with a default value (*noisy default*). Such deletion or substitution of cue information from experimental items could be caused by a variety of factors, including limited working memory capacity, background noise, inattention, and so on. Whatever the reason, the result is that the child inaccurately deploys this otherwise accurate cue information.

More specifically, both noisy deployment models rely on cue likelihoods ($P(f_{\text{CUE\_VAL}}|\alpha)$) accurately obtained from the input but access this information probabilistically via mixture modeling. The noisy deletion model (3) mixes the optimal model with models that delete one, two, or all three cues. In other words, when this modeled child is unable to deploy a given cue, she simply drops that cue's information.

$$\begin{aligned} P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}}) &\propto \\ [(\beta_{\text{MOR}})(\beta_{\text{CONN}})(\beta_{\text{FORM}})\ P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}}) &+ \\ (1-\beta_{\text{MOR}})(\beta_{\text{CONN}})(\beta_{\text{FOR}})\ P(\alpha|f_{\text{CONN}}, f_{\text{FORM}}) &+ \\ \ldots &+ \\ (1-\beta_{\text{MOR}})(1-\beta_{\text{CONN}})(1-\beta_{\text{FORM}})] &\times \\ P(\alpha) & \end{aligned} \quad (3)$$

The noisy default model (4) mixes the optimal model with models that substitute the cue's true value ([*acc*]) with a default ([*def*]), which we determined by sampling from the distribution of cue values in the child's input. In other words, when this modeled child is unable to deploy a given cue, she inserts a default value.

$$
\begin{aligned}
P(\alpha|f_{\text{MOR}}, f_{\text{CONN}}, f_{\text{FORM}}) \propto \\
[(\beta_{\text{MOR}})(\beta_{\text{CONN}})(\beta_{\text{FORM}}) \\
P(\alpha|f_{\text{MOR}} = [acc], f_{\text{CONN}} = [acc], f_{\text{FORM}} = [acc]) + \\
(1-\beta_{\text{MOR}})(\beta_{\text{CONN}})(\beta_{\text{FORM}}) \\
P(\alpha|f_{\text{MOR}} = [def], f_{\text{CONN}} = [acc], f_{\text{FORM}} = [acc]) + \\
\ldots + \\
(1-\beta_{\text{MOR}})(1-\beta_{\text{CONN}})(1-\beta_{\text{FORM}}) \\
P(\alpha|f_{\text{MOR}} = [def], f_{\text{CONN}} = [def], f_{\text{FORM}} = [def])] \times \\
P(\alpha)
\end{aligned} \quad (4)
$$

In both mixture models, the level of noise associated with each cue is determined by how much each sub-model contributes to the mix. Specifically, in (3) β_{MOR}, β_{CONN}, and β_{FORM} indicate the rate at which morphological, connective, and form cues are included, while in (4) they indicate the rate at which the accurate cue value is retained.

3.4 Results and discussion

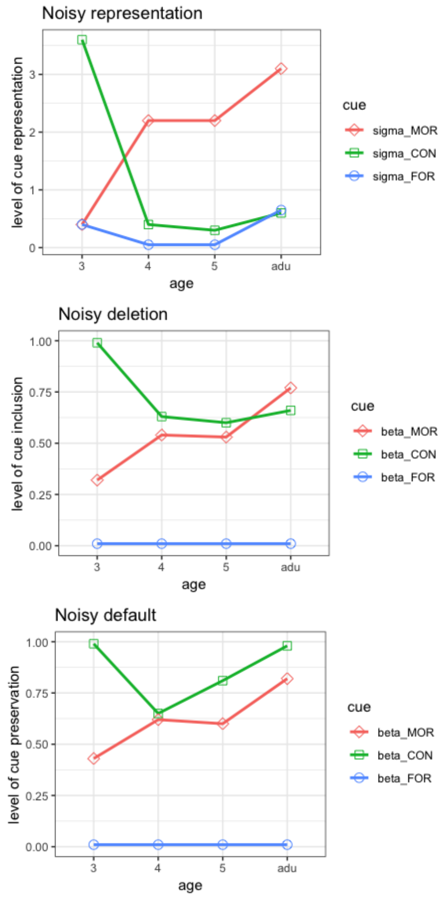
All three noisy models significantly outperform the baseline optimal model in capturing child behavior at age 3 (GLLR: all $\chi^2(3)>1641$, all $p<0.001$), 4 (all $\chi^2(3)>1451$, all $p<0.001$), and 5 (all $\chi^2(3)>976$, all $p<0.001$); notably, the same is true for adult behavior (all $\chi^2(3)>509$, all $p<0.001$). Among the three noisy models, the two noisy deployment models have lower overall MSEs and higher log likelihoods. This suggests that the noisy deployment models are consistently better at capturing the pronoun resolution behavior of the adults and children in this experiment and, moreover, that the underlying source of children's non-optimal pronoun interpretations is more likely to be an immature deployment of their otherwise mature representations of cue information.

In terms of the amount of noise associated with each cue, our models show similar developmental patterns (see Figure 2). First, there's a steady decrease in the noise associated with agreement morphology as children get older (i.e., the red lines show an increase in children's reliance on this cue). Second, the noise associated to connectives (green lines) is almost adult-like from age four on, while three-year-olds appear more sensitive than adults. Third, none of the best-fitting models indicate much use of pronominal form at all (i.e., blue lines close to 0)–including the best-fitting models for adults. This suggests that mostly ignoring pronominal form in this task is in fact adult-like.

4 Conclusion and future directions

Here we have shown how to use cognitive modeling to implement two different types of developmental theories about why children's interpretation of Spanish subject pronouns is non-adult-like. Our results suggest that immature deployment, rather than immature representation of cue information, is the more likely cause of children's behavior. In particular, younger children seem to inconsistently access their representations of how agreement morphology predicts the number of the pronoun's antecedent.

Figure 2: Best-fitting noise parameters for each model and age group. Larger values indicate less noise.



However, we do note that children show qualitatively different behavior in singular vs. plural conditions and that the noisy representation model is particularly bad at capturing this difference. This suggests that future work may improve model fit by using separate noise parameters for singular and plural morphology, rather than a single noise parameter for agreement morphology. This may result in a better quantitative fit to child behavior, especially for the noisy representation model.

More generally, our approach demonstrates how computational modeling can complement behavioral approaches to the investigation of language development, affording a clearer picture of what it is that changes as children grow into adults.

References

Hannah Forsythe. accepted. Resolving pronouns with multiple cues: Children use pragmatics before morphology. poster to be presented. In *44th annual Boston University Conference on Language Development (BUCLD44)*.

Michael C Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Annie Gagliardi, Naomi H Feldman, and Jeffrey Lidz. 2017. Modeling statistical insensitivity: Sources of suboptimal behavior. *Cognitive Science*, 41(1):188–217.

Noah D Goodman and Andreas Stuhlmüller. 2013. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in cognitive science*, 5(1):173–184.

Lisa Pearl and Benjamin Mis. 2016. The role of indirect positive evidence in syntactic acquisition: A look at anaphoric one. *Language*, 92(1):1–30.

Amy Perfors, Joshua Tenenbaum, Thomas Griffiths, and Fei Xu. 2011. A tutorial introduction ot bayesian models of cognitive development. *Cognition*, 120(3):302–321.

Gregory Scontras and Noah D Goodman. 2017. Resolving uncertainty in plural predication. *Cognition*, 168:294311.

Modelling Non-Local Maps as Strictly Piecewise Functions

Phillip Burness
University of Ottawa
pburn036@uottawa.ca

Kevin McMullin
University of Ottawa
kevin.mcmullin@uottawa.ca

A growing body of research aims to describe phonological patterns with *subregular* classes of formal languages or functions. Of particular note in this subregular hierarchy are the Strictly Local languages (SL; McNaughton and Papert, 1971; Rogers and Pullum, 2011; Rogers et al., 2013), and the Strictly Piecewise languages (SP; Heinz, 2010; Rogers et al., 2010, 2013), both of which are described in more detail below. The SL languages were extended to functions by Chandlee (2014) and Chandlee et al. (2014, 2015); the work presented here investigates whether the SP languages can be extended to functions in a similar way. While the SP functions may be able to capture non-local phonological processes that are not SL, it is somewhat difficult to achieve a straightforward definition of an SP function. As part of this work in progress, we first define a more powerful type of function and investigate whether the intended SP properties can be obtained by imposing specific restrictions.

The SL_k languages are those that ban certain contiguous sequences of length k, and have been put forth as a characterization of locally bounded phonotactic restrictions. A key property of SL_k languages is what is known as Suffix Substitution Closure (Rogers and Pullum, 2011; Rogers et al., 2013): any two well-formed strings in an SL_k language that share a suffix of length $k-1$ can both be legally continued by the same set of strings.

Chandlee (2014) and Chandlee et al. (2014, 2015) expanded on this property to define the Strictly Local functions, in which the the output associated with an input segment is determined by the immediately preceding $k-1$ elements on either the input side (ISL_k) or output side (OSL_k). These functions can model many local phonological processes such as substitution, deletion, and epenthesis. A major limitation of the SL languages and functions, though, is that they cannot model long-distance patterns such as sibilant harmony in Aari (e.g., /ʃed-er-s-it/ → [ʃederʃit], 'I was seen'; Hayward, 1990).

One proposed means of capturing long-distance patterns is to eliminate the requirement of contiguity. The SP_k languages operate in this manner, banning certain sequences of length k whether contiguous or not. For example, sibilant harmony in Aari can be described as a ban on output strings that contain the subsequence [ʃ...s]. As many non-local phonotactic dependencies can be characterized as SP languages (Heinz, 2010), this raises the question of whether the language class can be extended to functions as well. Preliminary work suggests that it may be possible to do so, though our line of inquiry faces an interesting challenge. Namely, the SP_k languages do not exhibit a property directly analogous to Suffix Substitution Closure, which makes it difficult to extend them to functions with the same approach that has been used for the ISL_k or OSL_k functions.

Interestingly, the class of Piecewise Testable languages (PT; Simon, 1975; Rogers et al., 2010, 2013), which are the boolean closure of the SP languages, *do* have such a suffix-related closure property. Rather than simply banning specific subsequences, a PT_k language is one that excludes strings with an impermissible *set* of subsequences of up to length k. Rogers et al.'s (2013) Theorem 7 states that given a PT_k language L and any two strings with a matching set of subsequences of up to length k, either both strings are in L or neither string is in L. A corollary of this Theorem is that any two well-formed strings in a PT_k language that have matching sets of subsequences of up to length k can both be legally continued by the same set of strings. The SP_k languages are effectively a restricted type of PT_k language, and we propose to define the SP_k functions as PT_k functions that satisfy certain restrictions.

Intuitively, an Output Piecewise k-Testable (OPT_k) function would keep track of the subsequences of up to length k produced so far, which would dictate the output for any subsequent input segment. For example, consider Figure 1 which shows how a hypothetical OPT_1 function would model the Aari sibilant harmony from above. Each circle represents the strings of length 1 (i.e., the individual segments) produced thus far, and an arrow labelled $x : y$ acts as instruction to output y and move to the indicated state upon reading x.

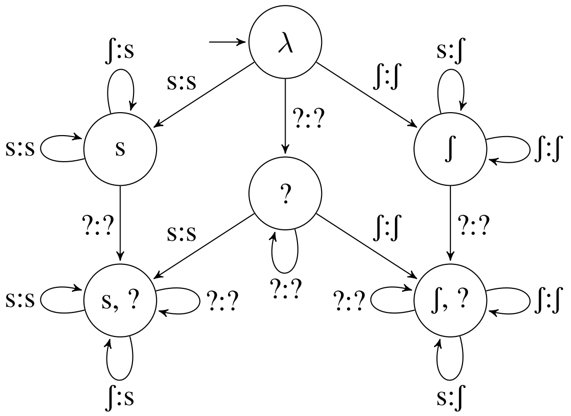


Figure 1: Aari sibilant harmony as an OPT_1 function, where ? denotes any non-sibilant

The key difference between an SP_k and a PT_k language is that a given k-subsequence has a consistent effect on well-formedness in an SP_k language; this is not necessarily true in a PT_k language. Since a PT_k language is defined with reference to *sets* of subsequences, it is possible for a PT_k language to exclude (i.e., treat as ill-formed) all strings containing a given k-length subsequence u, except those strings that also contain a different k-length subsequence v. Such conditional well-formedness of a k-length subsequence is impossible to describe using an SP_k language.

We therefore propose that the SP_k functions could be defined by restricting the PT_k functions such that the presence of a given subsequence has a consistent effect on the output for some input segment. A preliminary definition of the Output Strictly k-Piecewise (OSP_k) functions along these lines is provided below. $\mathtt{Sub}_{\leq k-1}(x)$ denotes the set of subsequences of up to length $k-1$ in a string x, and $\mathtt{cont}(\sigma, w)$ denotes the *contribution* of σ relative to w, or the output produced upon reading σ after having read w.

Definition 1. *A function $f : \Sigma^* \rightarrow \Delta^*$ is OSP_k iff:*

1. *If $\mathtt{cont}(\sigma, w)$ is undefined, then $\mathtt{cont}(\sigma, w')$ is undefined for all w' such that $\mathtt{Sub}_{\leq k-1}(f(w')) \supseteq \mathtt{Sub}_{\leq k-1}(f(w))$*
2. *If $\mathtt{Sub}_{\leq k-1}(f(w_1)) \neq \mathtt{Sub}_{\leq k-1}(f(w_2))$ and $\mathtt{cont}(\sigma, w_1) \neq \mathtt{cont}(\sigma, w_2)$, then either:*
   - *$\mathtt{cont}(\sigma, w_3) = \mathtt{cont}(\sigma, w_1)$ for all w_3 such that $\mathtt{Sub}_{\leq k-1}(f(w_3)) \supseteq (\mathtt{Sub}_{\leq k-1}(f(w_1)) \cup \mathtt{Sub}_{\leq k}(f(w_2)))$*
   - *$\mathtt{cont}(\sigma, w_3) = \mathtt{cont}(\sigma, w_2)$ for all w_3 such that $\mathtt{Sub}_{\leq k-1}(f(w_3)) \supseteq (\mathtt{Sub}_{\leq k-1}(f(w_1)) \cup \mathtt{Sub}_{\leq k}(f(w_2)))$*

The first point ensures that if some instance of an output subsequence causes the contribution of a following input element to be undefined in one case, then all instances of that output subsequence will cause the contribution of that following input element to be undefined. The second point ensures that when two subsequences have a different *defined* effect on the contribution of some input element, one of these is *dominant* and will apply to all mappings containing both subsequences. If this is indeed an appropriate definition of the OSP_k functions, an automata-theoretic characterization as in Figure 1 seems likely achievable.

Future work could then compare the OSP functions and the Output Tier-based Strictly Local functions (OTSL; Chandlee et al., 2017; Chandlee and McMullin, 2018; Burness and McMullin, 2019), which have also been put forth as a means of capturing non-local phonological maps. Like their name suggests, these are functional extensions of the Tier-based Strictly Local languages (TSL; Heinz et al., 2011; McMullin and Hansson, 2016). Rather than eschewing contiguity, the TSL languages and functions capture long-distance patterns by augmenting the SL languages and functions with a *tier*—a subset of the alphabet that allows us to ignore irrelevant elements that stand between interacting elements.

A decisive outcome from this comparison would have interesting consequences for phonological theory. If the OSP functions offer a better fit to the typology, it would suggest that local and non-local phonological maps are fundamentally different: the former operating according to strict precedence and the latter operating according to general precedence. If, on the other hand, the OTSL functions offer a better fit to the typology, it would suggest that all phonological maps operate according to strict precedence, relative to certain elements.

References

Phillip Burness and Kevin McMullin. 2019. Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90. Association for Computational Linguistics.

Jane Chandlee. 2014. *Strictly Local phonological processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*.

Jane Chandlee, Jeffrey Heinz, Adam Jardine, and Kevin McMullin. 2017. Modeling long-distance alternations with Tier-based Strictly Local functions. Paper presented at the 91st Annual Meeting of the Linguistic Society of America, Austin, TX.

Jane Chandlee and Kevin McMullin. 2018. Output-based computation and unbounded phonology. Presented at the *2018 Annual Meeting on Phonology*, San Diego, CA.

Richard J. Hayward. 1990. Notes on the Aari language. In Richard J. Hayward, editor, *Omotic language studies*, pages 425–493. School of Oriental and African Studies, University of London, London.

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, OR. Association for Computational Linguistics.

Kevin McMullin and Gunnar Ólafur Hansson. 2016. Long-distance phonotactics as Tier-based Strictly 2-Local languages. In *Proceedings of the 2014 Annual Meeting on Phonology*, Washington, DC. Linguistic Society of America.

Robert McNaughton and Seymour A. Papert. 1971. *Counter-free automata*. MIT Press, Cambridge, MA.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *The mathematics of language*, pages 255–265. Springer, Berlin.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3):329–342.

Imre Simon. 1975. Piecewise testable events. In *Automata Theory and Formal Languages: 2nd Grammatical Inference conference*, pages 214–222. Springer-Verlag.

Semantic categories of artifacts and animals reflect efficient coding

Noga Zaslavsky*
The Hebrew University
noga.zaslavsky@mail.huji.ac.il

Terry Regier
University of California, Berkeley
terry.regier@berkeley.edu

Naftali Tishby
The Hebrew University
tishby@cs.huji.ac.il

Charles Kemp
University of Melbourne
c.kemp@unimelb.edu.au

It has been argued that cross-language variation in semantic categories reflects pressure for efficient communication (e.g. Kemp et al. 2018). On this view, the lexicons of different languages represent a variety of means to the same functional end: transmitting ideas accurately, with minimal cognitive complexity. Recently, Zaslavsky et al. (2018) cast this idea in terms of an independent information-theoretic principle of efficiency, the Information Bottleneck (IB) principle (Tishby et al., 1999), which is closely related to Shannon's rate distortion theory. In this context, IB is given an underlying cognitive representation of a semantic domain, and a prior over objects in the domain, and it produces a set of optimal category systems for that domain, for different trade-offs between system complexity and accuracy. These optimal systems define the theoretical limit of efficiency. Zaslavsky et al. (2018) showed that IB explains much of the variation in color naming across languages, and also accounts for the emergence and evolution of named color categories, including soft structure and patterns of inconsistent naming. However, it has remained unclear to what extent this account generalizes to semantic domains other than color. Here we show that it generalizes to two qualitatively different semantic domains: names for containers, and for animals.

Containers. We considered container naming and pile-sorting data collected by White et al. (2017), relative to a stimulus set of 192 images of household containers (see Figure 2A for examples), produced by Dutch and French monolingual speakers, and by bilinguals in each of the two languages, yielding four conditions: language (Dutch, French) × linguistic status (monolingual, bilingual). We took the sorting data to provide domain structure, against which we assessed the efficiency of the naming data by applying the IB method. The results are shown in Figure 1. It can be seen that the efficiency of container naming in Dutch and French lies near the theoretical limit, for both monolinguals and bilinguals. These systems were also found to be more efficient than a set of hypothetical variants of these systems.

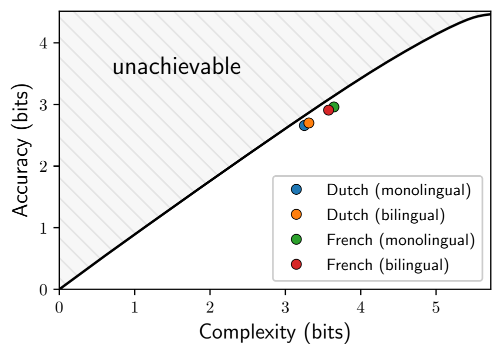


Figure 1: The black curve is the IB theoretical limit of efficiency for container naming, identifying the maximum achievable level of accuracy at each level of complexity. Points above this curve are unachievable. The four conditions considered exhibit near-optimal trade-offs between accuracy and complexity in container naming.

For visualization purposes, we embedded the 192 containers in a 2-dimensional space by applying non-metric multidimensional scaling (nMDS) with respect to the similarity data, similar to Ameel et al. (2009). We assigned a unique color to each container. The resulting 2D embedding and color coding of the containers stimulus set are shown in Figure 2A. The monolingual systems in Dutch and French are shown in Figure 2B, together with the corresponding optimal systems on the IB curve. It can be seen that the IB systems capture qualitative aspects of the actual systems, including soft categories and inconsistent naming

* Present address: Department of Brain and Cognitive Sciences, MIT.

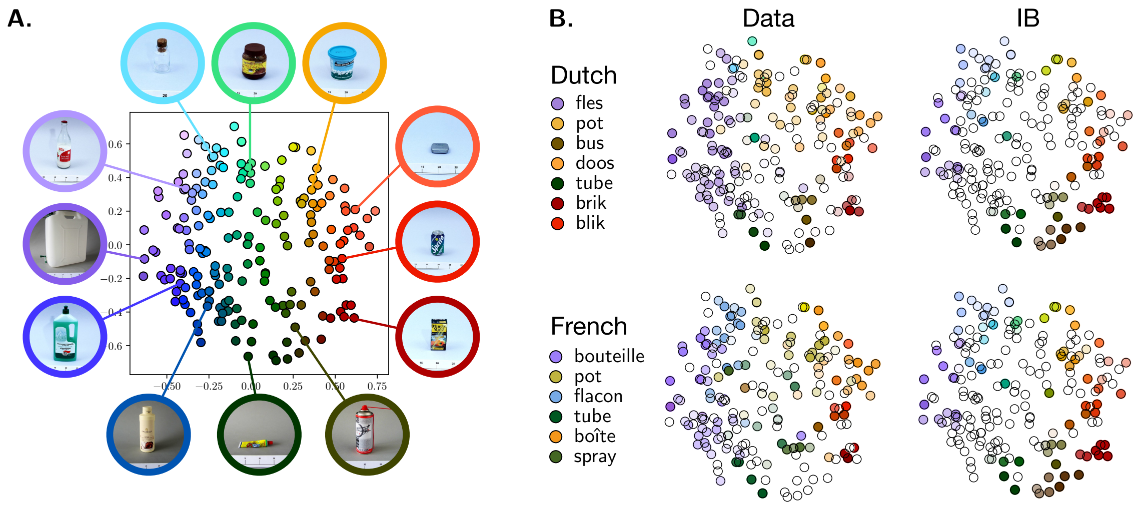


Figure 2: **A**. Two dimensional nMDS embedding and color coding of the containers stimulus set used by White et al. (2017). Images show a few examples. **B**. Monolingual naming distributions for Dutch (upper left) and French (lower left), together with their corresponding IB systems (right column), are visualized over the 2D embedding shown in (A). Each color corresponds to the color centroid of a container category, based on the color map in (A). Colors show category probabilities above 0.4, and color intensities reflect the values between 0.4 and 1. White dots correspond to containers for which no category is used with probability above 0.4. Legend for each language shows only major terms.

for some objects, showing that these phenomena can be explained by a drive for efficiency.

Animals. Brown (1984) proposed an implicational hierarchy of animal categories across languages, based on data from 144 languages. We conducted an analysis of animal naming broadly analogous to the container analysis described above, based on human-generated features and familiarity ratings drawn from the Leuven Natural Concept Database (De Deyne et al., 2008). That analysis (not illustrated or elaborated here for reasons of space) revealed that the IB theoretical limit of efficiency in this domain correctly predicts several aspects of Brown's hierarchy.

Conclusion. These findings suggest that fundamental information-theoretic principles of efficient coding may shape semantic categories across languages and across domains.

References

Eef Ameel, Barbara C. Malt, Gert Storms, and Fons Van Assche. 2009. Semantic convergence in the bilingual lexicon. *Journal of Memory and Language*, 60(2):270–290.

Cecil H. Brown. 1984. *Language and living things: Uniformities in folk classification and naming*. Rutgers University Press.

Simon De Deyne, Steven Verheyen, Eef Ameel, Wolf Vanpaemel, Matthew J. Dry, Wouter Voorspoels, and Gert Storms. 2008. Exemplar by feature applicability matrices and other Dutch normative data for semantic concepts. *Behavior Research Methods*, 40(4):1030–1048.

Charles Kemp, Yang Xu, and Terry Regier. 2018. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4(1).

Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. The Information Bottleneck method. In *37th Annual Allerton Conference on Communication, Control and Computing*.

Anne White, Barbara C. Malt, and Gert Storms. 2017. Convergence in the bilingual lexicon: A preregistered replication of previous studies. *Frontiers in Psychology*, 7.

Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *PNAS*, 115(31):7937–7942.