Evaluating Sequence Alignment for Learning Inflectional Morphology

David L. King
The Ohio State University
king.2138@osu.edu

Abstract

This work examines CRF-based sequence alignment models for learning natural language morphology. Although these systems have performed well for a limited number of languages, this work, as part of the SIGMORPHON 2016 shared task, specifically sets out to determine whether these models handle non-concatenative morphology as well as previous work might suggest. Results, however, indicate a strong preference for simpler, concatenative morphological systems.

Introduction

Morphologically-rich languages pose a challenge for the natural language processing and generation community. Computationally mapping inflected wordforms to a baseform has been standard practice in semantics and generation. Traditionally, hand-coding these as rule-based systems required extensive engineering overhead, but has produced high quality resolution between base and inflected wordforms. This work extends work by Durrett and DeNero (2013) to automatically learn morphological paradigms by comparing edit operations between a lemma and baseform and tests a similar algorithm on other morphologically-rich langauges and those which exhibit more extensive use of non-concatenative morphology.

Background

Morphological reinflection and lemma generation are not trivial tasks, and have been the subject of much research and engineering. Traditionally, rule-based and finite-state methods (Minnen et al., 2001; Koskenniemi, 1984) have been used, particularly when no training data is available. Although these handcrafted systems perform with a high level of accuracy, creating them is difficult and requires a great deal of engineering overhead.

Recently, more automatic, machine learning methods have been utilized. These systems have required far less handcrafting of rules, but also do not perform as well. Specifically, work by Durrett and DeNero (2013) exploits sequence alignment systems across strings, a technique originally developed for DNA analysis. They showed that by computing minimum edit operations between two strings and having a semi-markov conditional random field (CRF) (Sarawagi and Cohen, 2004) predict when wordform edits rules were to be used, a system could achieve state-of-the-art performance in completing morphological paradigms for English and German.

English and German, along with other Germanic languages, have a somewhat rarer tendency towards *ablauting*, that is changing or deleting segments from the lemma of a wordform as part of its inflection. In some circles, morphology is thought of in the purely *concatenative* sense (i.e. *give* + *-s* → *gives*). Durrent and DeNero's work shows promise in that they already account for non-concatenative morphonology in English and German. Using a similar system, this work hypothesizes that such an approach will perform well on languages with more prolific non-concatenative morphology, such as Arabic and Maltese.

Shared Task

The 2016 SIGMORPHON (Cotterell et al., 2016) shared task on morphological reinflection consisted of multiple tracks for discerning fully inflected wordforms in ten languages, two of which were surprise languages whose data was not released until a week before the submission deadline. In task 1, participants were given a lemma and a target word's grammatical information with

Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 49–53,
Berlin, Germany, August 11, 2016. ©2016 Association for Computational Linguistics

Language	Training Items
Arabic	12254
Finnish	12693
Georgian	11576
German	12490
Hungarian	16219
Maltese	18975
Navajo	6012
Russian	12390
Spanish	12575
Turkish	12336

Table 1: Training items available for restricted task 1.

which to guess the fully inflected target wordform. In task 2, participants were supplied with two fully inflected wordforms—one source and one target—and their grammatical features. Task 3 was the same as task 2, except that no source grammatical information was supplied.

Additionally, participants were allowed to choose standard, restricted, or bonus training sets. The standard training allowed for any task to use training data from a task lower than it. Restricted training only allowed for training on data for that given data set (i.e. task 1 can only train on task 1, task 2 on task 2, and task 3 on task 3). A system attempting a certain task number and training on a higher task number (e.g. attempting task 1 and additionally using task 2 training data) constituted using bonus training.

For the purposes of testing this work's hypothesis, task 1 was chosen as being the most analogous and direct means of evaluation. Additionally, restricted training was used to minimize variance between the training sets of the ten languages in question. As seen in table 1, although generally most training sets have about 12,000 items, Navajo, Maltese, and Hungarian are the exceptions.

Implementation

This work exploits string sequence alignment algorithms such as Hirschberg's algorithm (Hirschberg, 1975) and the Ratcliff/Obershelp algorithm (Black, 2004) in the same vein as recent work by Durrett and DeNero (2013) and Nicolai et al. (2015). In these frameworks, the fewest number of edits required to convert one string to another are considered to be morphological

give → gave

	g g	i a	v v	e e	
Rule		-i+a			

kitab → kutub

	k k	i u	t t	a u	b b	
Rule		-i+u		-a+u		

springen → gesprungen

		s s	p p	r r	i u	n n	g g	e e	n n	
	ge									
Rule	+g+e				-i +u					

Figure 1: Sample edits for English *give → gave*, Arabic *kitab to kutub* ('book' singular → plural), and German *springen → gesprungen* ('to jump' infinitival → past participle). Note that edit rules are applied in a character-by-character manner across the lemma.

rules. As shown in figure 1, source and target words are aligned to minimize edit operations required to make them the same. This minimal list of edit operations is converted into an edit rule at the character level (i.e. this work does not predict word level edit operations). These segment edits are fed with a feature set to be trained on by a linear chain CRF (Sutton and McCallum, 2011) using online passive-aggressive training (Crammer et al., 2006).

Features for the CRF included a mix of data provided by the task data and surface features from the uninflected lemmas. All features were shared across all segments (i.e. at the word level) except for features specific to the the current segment and listed in table 2. Outputs from the CRF were edit operations for each segment of the input lemma. After these operations were carried out on their respective segments within the lemma, a fully inflected wordform was the final output from the system. The feature set was chosen with insight from previous work.

Full feature set:

- Grammatical information – concatenated
- Grammatical information – factored
- Character level bigram information – forwards and backwards

Current	Edit	Affix type	Distance from beginning	Distance from end
start	+g+e	prefixing	0	8
s	empty	infixing	1	7
p	empty	infixing	2	6
r	empty	infixing	3	5
i	-u+i	infixing	4	4
n	empty	infixing	5	3
g	empty	infixing	6	2
e	empty	infixing	7	1
n	empty	suffixing	8	0

spring → gesprungen

Table 2: An example of character-specific features as used by the CRF – all other features are shared across the entire edit sequence.

- Character level trigram information – forwards and backwards
- Character indexes from beginning and end
- Distance from the current character to the beginning of the lemma
- Distance from the current character to the end of the lemma
- Affix type (prefixing, infixing, or suffixing – circumfixing was not explicitly encoded into the feature set)

Results

Overall the system performed far better on the development set than the test set. It is easiest to summarize the results from table 6 in terms of the number of edit rules the system had to learn. Languages with under 500 edit rules for the system to learn performed best and only experienced moderate dropoff between the development and test sets. Languages with over 500 edit rules to be learned both performed worse and experienced extreme drop offs in some instances. The exception, Turkish, will be discussed below and in the next section.

Languages traditionally used in these tasks, such as German, performed best, while those less often tested in these systems, such as Maltese, seem to be more difficult for the system to accurately predict. There was a drastic drop in Navajo, which the task organizers claim to be caused by a dialectal shift between the training, development,

Affix Data Set	Dev	Test
Train	-0.764	-0.707
Dev	-0.694	-0.603

Table 3: Correlations of the number of affixes per language in a given data set and the system's accuracy of that language.

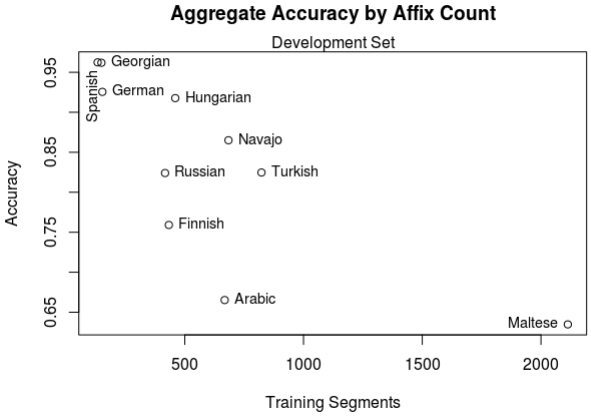

Figure 2: Aggregate Accuracy over the Development Set

and testing data sets, among other reasons. Maltese was not able to be tested since the CRF took 15 days to train, which did not fit within the time allotted for training on the surprise languages. The jump in training time for Maltese was not unexpected, given how many unique affixes the training set had, and taking into account the effect that increasing the number of classes a CRF must predict increases its asymptotic complexity quadratically by some measures (Cohn, 2007). Hungarian, the other surprise language, did not drop as drastically.

Language	Train	Dev
Arabic	3.249	3.170
Finnish	1.835	1.775
Georgian	1.464	1.474
German	1.042	1.035
Hungarian	1.559	1.536
Maltese	3.184	3.103
Navajo	3.260	3.283
Russian	1.803	1.775
Spanish	1.495	1.474
Turkish	2.131	2.058

Table 4: Entropy over affix counts in the training and development data sets.

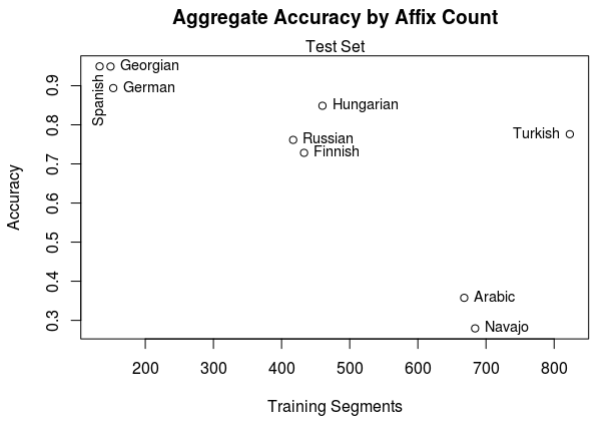

Figure 3: Aggregate Accuracy over the Test Set

Language	Train	Dev
Arabic	668	260
Finnish	433	228
Georgian	149	88
German	153	83
Hungarian	460	279
Maltese	2113	787
Navajo	684	386
Russian	417	170
Spanish	133	88
Turkish	823	438

Table 5: Number of unique affixes in each data set.

Discussion

The difference in system performance between development and testing could be interpreted as overfitting. That said, overfitting to the development set would show a more universal drop in scores from development to testing than is exhibited here. Table 5 shows the number of unique affixes the system had to learn. As expected, languages traditionally thought to have less complex morphological structure had fewer unique affixes in both training and development sets. This is echoed in table 4, where entropy over the unique affix counts was calculated.

In addition to a non-uniform drop in accuracy, a strong negative correlation–as seen in table 3–between the number of affixes in the training set and accuracy seems to indicate that data sparsity might explain this phenomenon more fully. It appears that data sparsity has a greater effect as the number of affixes increases.

Certain languages did appear to drop between development and testing more drastically than others. While Finnish, German, Hungarian, Russian, Spanish, and Turkish fell less that 10%, Navajo and Arabic fell more than 30% each. Navajo's drop can be explained by the lack of training data. In 6012 training items, there were 684 edit rules that the system had to learn. This ratio of edit rules to wordforms is more than 1:10, which is higher than almost any other language in the task, second only to Maltese. What is particularly interesting is the number of affixes between Turkish and Arabic.

Although Arabic fell more drastically, Turkish clearly has more affixes in the data set, both by ratio and sheer count, and should perform worse given the previously-mentioned observations about an overall negative correlation between affix number and system performance. It should be taken into consideration that the kinds of morphology in Arabic and Turkish are not entirely analogous. Turkish, although agglutinative, is also primarily a suffixing language (Lewis, 2000), while Modern Standard Arabic is comparatively more non-concatenative. Arabic and Maltese, both of which have high entropies as seen in table 4 in additional more non-concatenative morphological structures, also performed worse than Turkish in the development results, which had an entropy more akin to Russian and Finnish. This points to the likelihood that non-contatenative morphology is still an issue for sequence alignment algorithms. Whether this problem can be solved by using a different algorithm, increasing training data, or by altering the underlying machine learning is beyond the scope of this task.

It should also be noted that, as far as this work is aware, the data sets were not balanced for frequency. Language learners often rotely memorize irregular forms because they do not fit a productive inflectional pattern. Luckily, irregular forms usually occur more frequently than wordforms subject to productive morphological rules (Bybee and Thompson, 1997; Grabowski and Mindt, 1995). Since the algorithm ostensibly treats productive and lexicalized forms equally, it would be interesting to see if there were any difference in performance between these datasets and others balanced to account for irregular form frequency.

Conclusion

Sequence alignment algorithms have proven useful in automatically learning natural language

Language	Dev	Test
Arabic	0.665	0.358
Finnish	0.759	0.728
Georgian	0.962	0.949
German	0.925	0.894
Hungarian	0.918	0.849
Maltese	0.635	N/A
Navajo	0.865	0.279
Russian	0.824	0.761
Spanish	0.965	0.949
Turkish	0.825	0.776

Table 6: Aggregate Accuracy Across Languages. Maltese required 15 days to train, and was unable to finish before the results were due.

morphology. That said, supervised models require exceptional amounts of training data to overcome data sparsity. Given a lack of training data, more traditional finite-state methods might be preferable given enough time to engineer such systems. This work has shown that CRF-based sequence alignment models do perform well for languages with lower affix to wordform ratios and unique affix count entropy values. Although there is not enough evidence to overtly reject this work's hypothesis, the evidence does indicate a preference for concatenative morphology by CRF-based sequence alignment models.

Acknowledgments

This work acknowledges the contributions of Micha Elsner for advising and assisting both technically and theoretically, without which this would not have come together. This work also thanks the anonymous reviewers for their guidance and insight.

References

Paul E Black. 2004. Ratcliff/Obershelp pattern recognition. *Dictionary of Algorithms and Data Structures*, 17.

Joan Bybee and Sandra Thompson. 1997. Three frequency effects in syntax. In *Annual Meeting of the Berkeley Linguistics Society*, volume 23, pages 378–388.

Trevor A Cohn. 2007. *Scaling conditional random fields for natural language processing*. Ph.D. thesis, Citeseer.

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, Germany, August. Association for Computational Linguistics.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

Greg Durrett and John DeNero. 2013. Supervised Learning of Complete Morphological Paradigms. In *HLT-NAACL*, pages 1185–1195.

Eva Grabowski and Dieter Mindt. 1995. A corpus-based learning list of irregular verbs in English. *ICAME Journal*, 19(1995):5–22.

Daniel S. Hirschberg. 1975. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, pages 178–181. Association for Computational Linguistics.

G. Lewis. 2000. *Turkish Grammar*. Oxford: Oxford University Press.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(03):207–223.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proc. of NAACL*.

Sunita Sarawagi and William W Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192.

Charles Sutton and Andrew McCallum. 2011. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373.