Measuring the Value of Linguistics: A Case Study from St. Lawrence Island Yupik

Emily Chen
University of Illinois Urbana-Champaign / Urbana, IL
echen41@illinois.edu

Abstract

The adaptation of neural approaches to NLP is a landmark achievement that has called into question the utility of linguistics in the development of computational systems. This research proposal consequently explores this question in the context of a neural morphological analyzer for a polysynthetic language, St. Lawrence Island Yupik. It asks whether incorporating elements of Yupik linguistics into the implementation of the analyzer can improve performance, both in low-resource settings and in high-resource settings, where rich quantities of data are readily available.

1 Introduction

In the years to come, the advent of neural approaches will undoubtedly stand out as a pivotal point in the history of computational linguistics and natural language processing. The introduction of neural techniques has resulted in system implementations that are performant, but highly dependent on algorithms, statistics, and vast quantities of data. Still we consider this work to belong to computational linguistics, which raises the question: Where does *linguistics* fit in?

Researchers have endeavored to answer this question, though some years before the popularization of neural approaches, demonstrating in particular the value of linguistics to morphological and syntactic parsing (Johnson, 2011; Bender, 2013) as well as machine translation (Raskin, 1987). This question is all the more relevant now in light of machine learning; as such, the research proposed herein is an exploration of the value of linguistics and how its pairing with neural techniques consequently affects system performance.

2 Previous Work

As this question is too broad in scope to explore as is, we instead apply it to a specific context, and ask how the use of linguistics can facilitate the development of a neural morphological analyzer for the language St. Lawrence Island Yupik.

St. Lawrence Island Yupik, hereafter *Yupik*, is an endangered, polysynthetic language of the Bering Strait region that exhibits considerable morphological productivity. Yupik words may possess several derivational suffixes, such as **-pig** in (1), which are responsible for deriving new words from existing ones: **mangteghapig-** '*Yupik house*' from **mangteghagh-** '*house*'. Derivational suffixes are then followed by inflectional suffixes which mark grammatical properties such as case, person, and number.

(1) **mangteghapiput**
mangteghagh- -pig- -put
house- -real- ABS.PL.1PLPOSS
'*our Yupik houses*' (Nagai, 2001, p.22)

Analyzing a Yupik word into its constituent morphemes thus poses a challenge given the potential length and morphological complexity of that word, as well as the fact that its morphemes' actual forms may have been altered by the language's morphophonology (see § 4.2), as illustrated in (1). Moreover, since there exist few Yupik texts that could qualify as training data for a neural morphological analyzer, Yupik may also be considered a low-resource language.

Low-resource settings offer initial insights into how linguistics impacts a morphological analyzer's performance. While many neural systems perform well when they are trained on a multitude of data points, studies have shown that utilizing linguistic concepts and incorporating language features can enhance performance in settings where training data is scarce.

With respect to the task of morphological analysis in particular, Moeller et al. (2019) demonstrated that when data was limited to 10,000 to

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 27–33
Florence, Italy, July 28 - August 2, 2019.

30,000 training examples, a neural morphological analyzer for Arapaho verbs that considered linguistically-motivated intermediate forms ultimately outperformed the analyzer that did not.

3 Linguistics in Low-Resource Settings

Given the success of Moeller et al. (2019)'s study, we replicated the morphological parsing or analysis experiments for Yupik nouns, studying the extendability of the claim that incorporating linguistics eases the task of morphological analysis.

3.1 Methodology

3.1.1 Morphological Analysis as Machine Translation

Initial steps toward recreating the Arapaho experiments involved recasting morphological analysis as a sequence-to-sequence machine translation task. The input sequence consists of characters that comprise the surface form, such as **whales**, which is *translated* into an output sequence of characters and morphological tags that comprise the glossed form, such as **whale[PL]**:

w h a l e s
↓
w h a l e [PL]

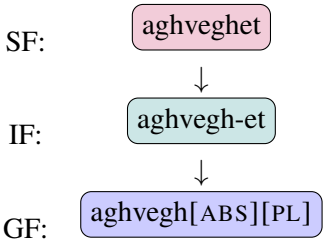
The morphological analysis of the Yupik surface form in (2) can consequently be regarded as the following translation:

a g h v e g h e t
↓
a g h v e g h [ABS] [PL]

Observe that the glossed form resembles the interlinear morphological gloss, underlined in (2), which offers a lexical or linguistic description of each individual morpheme.

(2) **aghveghet**
aghvegh- -et
whale- -ABS.PL
'*whales*'

While this methodology of training a machine translation system to translate between surface forms and glossed forms (the *direct strategy*) has resulted in fairly successful morphological analyzers (Micher, 2017; Moeller et al., 2018; Schwartz et al., 2019), Moeller et al. (2019) found that supplementing the training procedure with an intermediate translation step (the *intermediate strategy*) improved the performance of the Arapaho verb analyzer in instances of data scarcity. This intermediate step utilized the second line seen in (2) that is neglected in the direct strategy, but is regarded as significant by linguists for listing constituent morphemes in their full forms. As a result, in addition to training an analyzer via the direct strategy, Moeller et al. (2019) trained a second analyzer via the intermediate strategy, that performed two sequential translation tasks, from *surface form* (SF) to *intermediate form* (IF), and from intermediate form to *glossed form* (GF).



3.1.2 Generating Training Data

The training data in our replicated study consequently consisted of Yupik SF-IF-GF triplets. Like the training sets described in Moeller et al. (2019), the Yupik datasets were generated via the existing finite-state morphological analyzer (Chen and Schwartz, 2018), implemented in the `foma` finite-state toolkit (Hulden, 2009). Since analyzers implemented in `foma` perform both morphological analysis (SF→GF) and generation (GF→SF) and permit access to intermediate forms, the glossed forms were generated first, by pairing a Yupik noun root with a random selection of derivational suffixes, and a nominal case ending, as in (3) (see § 4.1 for a more detailed discussion).

(3) aghvegh-ghllag[ABS][PL]

Each glossed form's intermediate and surface forms were subsequently generated via our Yupik finite-state analyzer (Chen and Schwartz, 2018), resulting in triplets such as the one seen below:

SF	aghveghllaget
IF	aghvegh-ghllag-et
GF	aghvegh-ghllag[ABS][PL]

Each triplet was split into three training sets, consisting of the following parallel data:

1. SF → IF
2. IF → GF
3. SF → GF

The first two sets were used to train the analyzer via the intermediate strategy, and the last set was used to train the analyzer that adhered to the direct strategy. Lastly, whereas Moeller et al. (2019) developed training sets consisting of 14.5K, 18K, 27K, 31.5K, and 36K examples, the Yupik training sets varied from 1K to 20K examples in increments of 5000, to more realistically represent the low-resource setting of Yupik.

3.1.3 Training Parameters

For training, each parallel dataset was tokenized by character and randomly partitioned into a training set, a validation set, and a test set in a 0.8 / 0.1 / 0.1 ratio. The two analyzers trained on each of these datasets were then implemented as bidirectional recurrent encoder-decoder models with attention (Schuster and Paliwal, 1997; Bahdanau et al., 2014) in the Marian Neural Machine Translation framework (Junczys-Dowmunt et al., 2018). We used the default parameters of Marian, described in Sennrich et al. (2016), where the encoder and decoder consisted of one hidden layer each, and the model was trained to convergence via early stopping and holdout cross validation.

3.2 Results

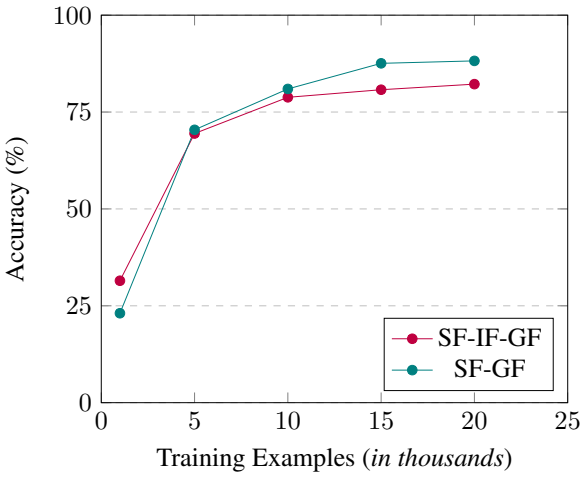


Figure 1: Accuracy scores of the analyzers trained on the intermediate and direct strategies, for all five datasets

The two trained analyzers for each dataset were evaluated on identical held-out test sets in order to compare their performances. As illustrated in Figure 1, it was only in the lowest data setting that the intermediate strategy outperformed the direct strategy with respect to accuracy. In all other instances, the direct strategy emerged as the better training methodology.

We speculate that this disparity in our results and that of Moeller et al. (2019) is due to differences in the morphophonological systems of Arapaho and Yupik and their effects on spelling. Arapaho's morphophonology, in particular, can radically alter the spelling of morphemes in the GF versus SF of a given word, as seen below (Moeller et al., 2019). It is possible that the intermediate step consequently assists the Arapaho analyzer in bridging this orthographical gap.

SF	nonoohobeen
IF	noohoween
GF	[VERB][TA][ANIMATE-OBJECT] [AFFIRMATIVE][PRESENT] [IC]noohow[1PL-EXCL-SUBJ][2SG-OBJ]

In Yupik, however, there is considerably less variation in the spelling (see § 3.1.2). This may mean the addition of the intermediate step in the Yupik analyzer only creates more room for error, and the direct strategy fares better as a result.

Though the results of our replicated study seem to point to the expendability of linguistics for the task of morphological analysis, calculating the Levenshtein distances between the incorrect outputs of each analyzer and their gold standard outputs offers a novel interpretation.

For every morphological analysis flagged as incorrect, its Levenshtein distance to the correct analysis was calculated, and all such distances were averaged for each analyzer (see Figure 2).

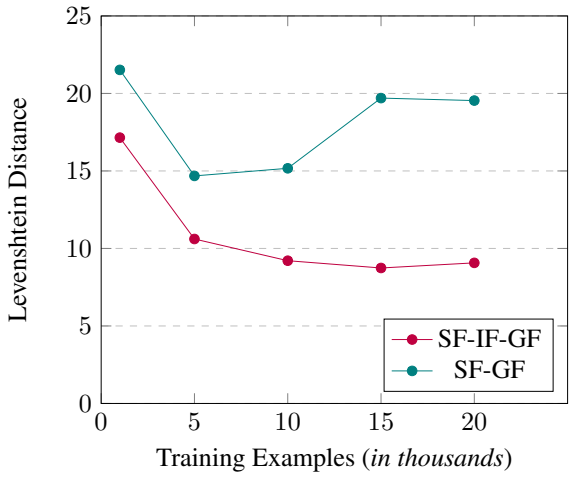


Figure 2: Average Levenshtein distances of the analyzers trained on the intermediate and direct strategies, for all five datasets

(4) **nunivagseghat**
nunivagseghagh- -t
tundra.vegetation- -ABS.PL
'*tundra vegetation*' (Nagai, 2001, p.60)

(5) **Sivuqaghhmiinguunga**
Sivuqagh- -mii- -ngu- -u- -nga
St. Lawrence Island- -resident.of- -to.be. -INTR.IND- -1SG
'*I am a St. Lawrence Islander*' (Jacobson, 2001, p.42)

(6) **ilughaghniighunneghtughllagyalghiit**
ilughagh- -niigh- -u- -negh- -tu- -ghllag- -yalghii- -t
cousin- -tease- -do- -very.many- -do.habitually- -very- -INTR.PTCP_OBL- -3PL
'*Many cousins used to teach each other a lot*' (Apassingok et al., 1993, p.47)

We found that the average Levenshtein distance for the analyzer trained on the intermediate strategy was statistically less than that of the direct strategy analyzer ($p < 0.0001$), with the exception of the lowest data setting. At 15K and 20K training examples, for instance, the average Levenshtein distances differed by nearly 10 or 11 operations. Furthermore, there did not appear to be a statistically significant difference in the complexity of the analyses being flagged as incorrect; the direct strategy was just as likely as the intermediate strategy to misanalyze simple words with one or two derivational suffixes.

The shorter Levenshtein distances suggest that the analyzers trained on the intermediate strategy consistently returned analyses that better resembled the correct answers as compared to their direct strategy counterparts. Therefore, even though the direct strategy proved superior to the intermediate strategy with respect to general accuracy, the outputs of the intermediate strategy may be more valuable to students of Yupik who are more reliant on the neural analyzer for an initial parse.

4 Linguistics in High-Resource Settings

The replicated study suggests that the accuracy of the analyzer is proportional to the quantity of training examples, especially for the direct strategy, as evidenced in Figure 1. Additional experiments demonstrated, however, that even using the finite-state analyzer to generate as many as 10 million training examples resulted in the accuracy of the neural analyzer plateauing around 88.77% for types and 87.19% for tokens on a blind test set that encompassed 659 types and 796 tokens respectively. This raises the question as to whether it is possible to improve the neural analyzer to competitive accuracy scores above 90% by reinforcing the direct strategy with aspects of Yupik linguistics whose effects have yet to be explored. Thus, the remainder of this proposal introduces these linguistic aspects and suggests means of integrating them into the high-resource implementation of the neural analyzer.

4.1 Integrating Yupik Morphology

One aspect of Yupik that may be useful is its word structure, which typically adheres to the following template, where ( ) denotes optionality:

Root + (*Derivational Suffix(es)*) + *Inflectional Suffix(es)* + (*Enclitic*)

Most roots can be identified as common nouns or verbs and are responsible for the most morphologically complex words in the language, as they are the only roots that can take derivational suffixes. Moreover, all derivational morphology is suffixing in nature, and Yupik words may have anywhere from zero to seven derivational suffixes, with seven being the maximum that has been attested in Yupik literature (de Reuse, 1994). Lastly, there are two types of inflection in Yupik: nominal inflection and verbal inflection.

This word structure consequently results in Yupik words of varying length as well as varying morphological complexity (see (4), (5), and (6)), which in turn constitutes ideal conditions for *curriculum learning*.

Curriculum learning, with respect to machine learning, is a training strategy that "introduces dif-

ferent concepts at different times, exploiting previously learned concepts to ease the learning of new abstractions" (Bengio et al., 2013). As such, "simple" examples are presented in the initial phases of training, with each phase introducing examples that are progressively more complex than the last, until the system has been trained on all phases, that is, the full *curriculum*.

The morphological diversity of Yupik words is naturally suited for curriculum learning, and may positively impact the accuracy of the neural analyzer. One proposed experiment of this paper is to restructure the training dataset, such that the neural analyzer is trained on the simplest Yupik words first, that is, those words consisting of an inflected root with zero derivational suffixes. Each successive phase introduces words with an additional derivational suffix, until the last phase presents the most morphologically complex words attested in the language.

4.2 Integrating Yupik Morphophonology

A second aspect of Yupik linguistics that may be integrated is its complex morphophonological rule system. In particular, the suffixation of derivational and inflectional morphemes in Yupik is conditioned by morphophonological rules that apply at each morpheme boundary and obscure them, rendering a surface form that may be unrecognizable from the glossed form, as in (7):

(7) **kaanneghituq**
kaate- -nghite- -u- -q
arrive- -did.not- -INTR.IND- -3SG
'*he/she did not arrive*' (Jacobson, 2001, p.43)

Moreover, each morphophonological rule has been assigned an arbitrary symbol in the Yupik literature (Jacobson, 2001), and so, every derivational and inflectional suffix can be written with all of the rules associated with it, as in (8). Here, @ modifies root-final *-te*, **–** deletes root-final consonants, $\sim_{\mathbf{f}}$ deletes root-final *-e*, and **(g/t)** designates allomorphs that surface under distinct phonological conditions.

(8) **kaanneghituq**
kaate- -@–nghite- -$\sim_{\mathrm{f}}$(g/t)u- -q
arrive- -did.not- -INTR.IND- -3SG
'*he/she did not arrive*' (Jacobson, 2001, p.43)

A second proposed experiment will consequently explore the potential insight provided by including these morphophonological symbols in the training examples, studying whether the symbols facilitate learning of the surface form to glossed form mapping or whether these additional characters actually introduce noise. Since minimal pairs do exist to differentiate the phonological conditions under which each symbol applies (see (9)), inclusion of the symbols may in fact assist the system in learning the morphophonological changes that are induced by certain suffixes.

(9)
nuna–ghllak	→	nunaghllak
qulmesiite–ghllak	→	qulmesiiteghllak
anyagh–ghllak	→	angyaghllak
sikig–ghllak	→	sikigllak
kiiw–ghllak	→	kiiwhllagek

Lastly, Yupik morphophonology may also be integrated into a curriculum learning training strategy, where separating the "easy-to-learn" training examples from the "hard-to-learn" training examples can be accomplished in the following ways:

1. Quantifying the number of morphophonological rules associated with a given morpheme, such that the simplest training examples encompass all suffixes with zero symbols attached, such as **-ni** '*the smell of; the odor of; the taste of*' (Badten et al., 2008, p.658). Subsequent phases successively increase this quantity by one.

2. Ranking the morphophonological rules themselves by difficulty, such that the initial phase introduces Yupik suffixes with the rules that have been deemed "easiest to learn", while future phases gradually introduce those that are "harder to learn" [1].

5 Presenting A Holistic Experiment

In summary, the objective of this proposed research is to utilize aspects of the Yupik language to reinforce the direct strategy in high-resource settings, guiding how the training examples are structured and the nature of their content. Previous sections share possible ways in which these linguistic elements of Yupik may be taken into account, but they can in fact be integrated into a single holistic experiment that trains multiple analyzers with varying degrees of linguistic information.

[1] A difficulty ranking was elicited from a single student during fieldwork conducted in March 2019, as most Yupik students had not yet mastered the symbols and the rules they represented.

In particular, we propose developing several sets of training data with the following characteristics:

1. Includes the morphophonological symbols (§ 4.2)
2. Ranks the training examples with respect to the number of morphemes (§ 4.1)
3. Ranks the training examples with respect to the number of morphophonological symbols per morpheme (§ 4.2)
4. Ranks the training examples with respect to the learning difficulty of the symbols (§ 4.2)

Each training dataset will incorporate as many or as few of these characteristics as desired, for a total of 15 datasets ($\binom{4}{4} + \binom{4}{3} + \binom{4}{2} + \binom{4}{1}$), and by extension, 15 neural analyzers. We expect any training set that involves morphophonological symbols to improve upon the existing analyzer's ability to distinguish between otherwise homographic suffixes, often a point of confusion. Taking morpheme count into consideration may also improve the analyzer's handling of words with relatively few derivational suffixes ($\sim$0-3), leaving the bulk of errors to instead comprise the most morphologically complex words. Furthermore, by virtue of training on an organized dataset rather than a randomly selected one, we predict that the analyzer will be exposed to a much more equal distribution of Yupik roots and suffixes. It should then be less likely than it is now to invent roots and suffixes that conform morphophonologically, but do not actually exist in the attested lexicon. Lastly, the performance of these analyzers can be compared to the performance of a baseline system, that is simply trained on the direct strategy without any morphophonological symbols or structure to its training data.

6 Conclusion

Moeller et al. (2019) and the replicated study for Yupik presented herein suggest that the use of linguistics can positively impact the performances of neural morphological analyzers, at least in lower resource settings. The proposed research, however, seeks to extend this observation to any data setting, and explore the effects of incorporating varying degrees of linguistic information in the training data, in hopes of shedding light on how best to approach to the task of morphological analysis via machine learning.

Acknowledgments

Portions of this work were funded by NSF Documenting Endangered Languages Grant #BCS 1761680, and a University of Illinois Graduate College *Illinois Distinguished Fellowship*. Special thanks to the Yupik speakers who have shared their language and culture with us.

References

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. 1993. *Kallagneghet / Drumbeats*. Bering Strait School District, Unalakleet, Alaska.

Linda Womkon Badten, Vera Oovi Kaneshiro, Marie Oovi, and Christopher Koonooka. 2008. *St. Lawrence Island / Siberian Yupik Eskimo Dictionary*. Alaska Native Language Center, University of Alaska Fairbanks.

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

Emily M. Bender. 2013. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Morgan & Claypool Publishers.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Emily Chen and Lane Schwartz. 2018. A morphological analyzer for St. Lawrence Island / Central Siberian Yupik. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan.

Mans Hulden. 2009. Foma: A finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32. Association for Computational Linguistics.

Steven A. Jacobson. 2001. *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language, Preliminary Edition*, 2nd edition. Alaska Native Language Center, Fairbanks, Alaska.

Mark Johnson. 2011. How relevant is linguistics to computational linguistics? *Linguistic Issues in Language Technology*, 6.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T.

Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Jeffrey Micher. 2017. Improving coverage of an inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages pp. 101–106, Honolulu. Association for Computational Linguistics.

Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2018. A neural morphological analyzer for Arapaho verbs learned from a finite state transducer. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, Santa Fe, New Mexico. Association for Computational Linguistics.

Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2019. Improving low-resource morphological learning with intermediate forms from finite state transducers. *Proceedings of the Workshop on Computational Methods for Endangered Languages: Vol. 1*.

Kayo Nagai. 2001. *Mrs. Della Waghiyi's St. Lawrence Island Yupik Texts with Grammatical Analysis*. Number A2-006 in Endangered Languages of the Pacific Rim. Nakanishi Printing, Kyoto, Japan.

Victor Raskin. 1987. Linguistics and natural language processing. *Machine Translation: Theoretical and Methodological Issues*, pages 42–58.

Willem J. de Reuse. 1994. *Siberian Yupik Eskimo — The Language and Its Contacts with Chukchi*. Studies in Indigenous Languages of the Americas. University of Utah Press, Salt Lake City, Utah.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Lane Schwartz, Emily Chen, Sylvia Schreiner, and Benjamin Hunt. 2019. Bootstrapping a neural morphological analyzer for St. Lawrence Island Yupik from a finite-state transducer. *Proceedings of the Workshop on Computational Methods for Endangered Languages: Vol. 1*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. *arXiv preprint arXiv:1606.02891*.