Bootstrapping a Neural Morphological Analyzer for St. Lawrence Island Yupik from a Finite-State Transducer

Lane Schwartz
University of Illinois
at Urbana-Champaign
lanes@illinois.edu

Emily Chen
University of Illinois
at Urbana-Champaign
echen41@illinois.edu

Benjamin Hunt
George Mason University
bhunt6@gmu.edu

Sylvia L.R. Schreiner
George Mason University
sschrei2@gmu.edu

Abstract

Morphological analysis is a critical enabling technology for polysynthetic languages. We present a neural morphological analyzer for case-inflected nouns in St. Lawrence Island Yupik, an endangered polysythetic language in the Inuit-Yupik language family, treating morphological analysis as a recurrent neural sequence-to-sequence task. By utilizing an existing finite-state morphological analyzer to create training data, we improve analysis coverage on attested Yupik word types from approximately 75% for the existing finite-state analyzer to 100% for the neural analyzer. At the same time, we achieve a substantially higher level of accuracy on a held-out testing set, from 78.9% accuracy for the finite-state analyzer to 92.2% accuracy for our neural analyzer.

1 Introduction

St. Lawrence Island Yupik, henceforth Yupik, is an endangered polysynthetic language spoken on St. Lawrence Island, Alaska and the Chukotka Peninsula of Russia. Members of the Yupik community on St. Lawrence Island have expressed interest in language revitalization and conservation.

Recent work by Chen and Schwartz (2018) resulted in a finite-state morphological analyzer for Yupik implemented in `foma` (Hulden, 2009). That analyzer implements the grammatical and morphophonological rules documented in *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language* (Jacobson, 2001).

In this work, we test the coverage of the finite-state analyzer against a corpus of digitized Yupik texts and find that the analyzer fails to return any analysis for approximately 25% of word types (see §2 and Table 1). We present a higher-coverage neural morphological analyzer for case-inflected Yupik nouns that involve no derivational morphology, using the previously-developed finite-state analyzer to generate large amounts of labeled training data (§3). We evaluate the performance of the finite-state and neural analyzers, and find that the neural analyzer results in higher coverage and higher accuracy (§5), even when the finite-state analyzer is augmented with a guessing module to hypothesize analyzes for out-of-vocabulary words (§4). We thus find that a robust high-accuracy morphological analyzer can be successfully bootstrapped from an existing lower-coverage finite-state morphological analyzer (§6), a result which has implications for the development of language technologies for Yupik and other morphologically-rich languages.

2 Evaluation of the FST Analyzer

The finite-state morphological analyzer of Chen and Schwartz (2018) implements the grammatical and morphophonological rules documented in *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language* (Jacobson, 2001) using the `foma` finite-state toolkit (Hulden, 2009).

In order to evaluate the percentage of attested Yupik word forms for which the finite-state analyzer produces any analysis, we began by digitizing several hundred Yupik sentences presented in Jacobson (2001) as examples to be translated by the reader. We next assembled, digitized, and manually validated seven texts that each consist of a collection of Yupik stories along with corresponding English translations. The texts include four anthologies of Yupik stories, legends, and folk tales, along with three leveled elementary primers prepared by the Bering Strait School District in the 1990s (Apassingok et al., 1993, 1994, 1995). Of the four anthologies, three comprise a trilogy known as *The Lore of St. Lawrence Is-*

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages, pages 87–96,
Honolulu, Hawai'i, USA, February 26–27, 2019.

Text	% Coverage		Corpus size
	Tokens; Types		(in words)
Ref	98.24	97.87	795
SLI1	79.10	70.62	6859
SLI2	77.14	68.87	11,926
SLI3	76.98	68.32	12,982
Ungi	84.08	73.45	15,766
Lvl1	76.64	70.86	4357
Lvl2	75.42	72.62	5358
Lvl3	77.71	75.19	5731

Table 1: For each Yupik text, the percentage of types and tokens for which the Yupik finite-state analyzer of Chen and Schwartz (2018) returns an analysis, along with the total number of tokens per text. *Ref* refers to Yupik examples taken from the Jacobson (2001) reference grammar, *SLI1* - *SLI3* refer to the *Lore of St. Lawrence Island*, volumes 1-3 (Apassingok et al., 1985, 1987, 1989), *Ungi* is an abbreviation for *Ungipaghaghlanga* (Koonooka, 2003), and *Lvl1* - *Lvl3* refer to the elementary Yupik primers (Apassingok et al., 1993, 1994, 1995).

land (Apassingok et al., 1985, 1987, 1989), while the last is a stand-alone text, *Ungipaghaghlanga* (Koonooka, 2003; Menovshchikov, 1988). Together, these texts represent the largest known collection of written Yupik.

After digitizing each text, we analyzed each Yupik word in that text using the finite-state morphological analyzer. We then calculated the percentage of tokens from each text for which the finite-state analyzer produced at least one analysis. We call this number *coverage*, and report this result for each text in Table 1. The mean coverage over the entire set of texts was 77.56%. The neural morphological analyzer described in the subsequent section was developed in large part to provide morphological analyses for the remaining 22.44% of heretofore unanalyzed Yupik tokens.

.

3 Yupik Morphological Analysis as Machine Translation

The task of morphological analysis can be regarded as a machine translation (sequence-to-sequence) problem, where an input sequence that consists of characters or graphemes in the source language (surface form) is mapped to an output sequence that consists of characters or graphemes *and* inflectional tags (underlying form). For example, in English, the sequence of characters representing the surface word form *foxes* can be transformed into a sequence of characters representing the root word *fox* and the inflectional tag indicating plurality:

f o x e s
↓
f o x [PL]

In contrast to English, Yupik is highly productive with respect to derivational and inflectional morphology.[1] See §3.1.1 for noun inflection tags.

(1) **kaviighet**
kaviigh- -∼$_{sf}$-w:(e)t
fox- -ABS.UNPD.PL
'*foxes*'

But in much the same way, (1) can be rewritten as a translation process from an input sequence of graphemes that represent the surface form to an output sequence of graphemes and inflectional tags that represent the underlying form:

k a v i i gh e t
↓
k a v i i gh [ABS] [UNPD] [PL]

3.1 Generating Data from an FST

Very little Yupik data has previously been manually annotated in the form of interlinear glosses. On the other hand, the finite-state morphological analyzer of Chen and Schwartz (2018) is capable of generating Yupik surface forms from provided underlying forms, and vice versa. In the following sections, bracketed items [..] introduce the inflectional tags that are used in the underlying forms of the finite-state analyzer.

3.1.1 Basic Yupik Nouns

Yupik nouns inflect for one of seven grammatical cases:

1. *ablative-modalis* [ABL_MOD]
2. *absolutive* [ABS]
3. *equalis* [EQU]
4. *localis* [LOC]
5. *terminalis* [TER]
6. *vialis* [VIA]
7. *relative* [REL]

[1] Each derivational and inflectional suffix is associated with a series of morphophonological rules. Each rule is represented by a unique symbol such as – or ∼$_{sf}$ as introduced in Jacobson (2001). This convention is used in the Yupik grammar, in the Yupik-English dictionary (Badten et al., 2008), and by Chen and Schwartz (2018). We therefore follow this convention, employing these symbols in our glosses here. See Table 2 on the following page for more details.

Symbol	Description
∼	Drops *e* in penultimate (semi-final) position or *e* in root-final position and *hops* it *e-Hopping* is the process by which vowels *i*, *a*, or *u* in the first syllable of the root are lengthened as a result of dropping semi-final or final-*e*, so termed because it is as if the *e* has "hopped" into the first syllable and assimilated. *e-Hopping* will not occur if doing so results in a three-consonant cluster within the word or a two-consonant cluster at the beginning (Jacobson, 2001).
$\sim_{f}$	Drops final *e* and hops it
$\sim_{sf}$	Drops semi-final *e* and hops it
-w	Drops weak final consonants, that is, *gh* that is not marked with an *. Strong *gh* is denoted *gh**
:	Drops uvulars that appear between single vowels
–	Drops final consonants
—	Drops final consonants and preceding vowel
@	Indicates some degree of modification to root-final *te*, the degree of which is dependent on the suffix
+	Indicates no morphophonology occurs during affixation. This symbol is implicitly assumed if no other symbols are present.

Table 2: List of documented morphophonological rules in Yupik and their lexicalized symbols. For more details see Jacobson (2001) and Badten et al. (2008).

Nouns then remain *unpossessed* [UNPD] and inflect for number (*singular* [SG], *plural* [PL], or *dual* [DU]), or inflect as possessed nouns. Possessed nouns are marked for number, and for the person and number of the possessor. For example, [1SGPOSS][SGPOSD] marks a possessed singular noun with a first person singular possessor.

The Badten et al. (2008) Yupik-English dictionary lists 3873 noun roots, and the Jacobson (2001) reference grammar lists 273 nominal inflectional suffixes. We deterministically generated data by exhaustively pairing every Yupik noun root with every inflectional suffix (ignoring any semantic infelicity that may result). As shown in Table 3 above, the underlying forms that result from these pairings map to just over 1 million inflected Yupik word forms or surface forms. Note that these word forms represent morphologically licit forms, but not all are attested. Even so, we did not exclude unattested forms nor weight them according to plausibility, since we lack sufficient documentation to distinguish the valid forms from the semantically illicit ones. This parallel dataset of inflected Yupik nouns and their underlying forms represents our corpus.

Root		Case		Poss		Posd		*Total*
3873	×	7	×	1	×	3	=	81,333
3873	×	7	×	12	×	3	=	975,996
								1,057,329

Table 3: Extracted training data. The first row counts the total number of *unpossessed* nouns which are marked for number: [SG], [PL], [DU]. The second row counts the total number of *possessed* nouns which are marked for number and also for 12 differing types of possessors, which themselves are marked for person, [1-4], and number, [SG], [PL], [DU].

3.1.2 Identified Flaw in Existing Yupik FST

While generating data using the finite-state analyzer, we observed a minor bug. Specifically, the finite-state analyzer fails to account for the allomorphy that is triggered on some noun roots when they are inflected for a subset of the 3rd person possessor forms (root-final *-e* surfaces as *-a*).

(2) **neqangit**
neqe- -∼:(ng)it
food- -ABS.PL.3PLPOSS
'*their foods*'

As shown in (2), the correct surface form for **neqe**[ABS][PL][3PLPOSS] is **neqangit**, but the analyzer incorrectly generates **neqngit** instead. Due to time constraints and the relatively small estimated impact, we did not modify the analyzer to correct this bug. Though this impacts the training data, having previously evaluated the analyzer, we do not believe this error to be egregious enough to compromise the generated dataset.

3.1.3 Yupik Verbs and Derivational Suffixes

While our training set for this work is the set of inflected Yupik nouns described in §3.1.1, it is important to note that this process could in principle be used to generate a much larger training set. The Badten et al. (2008) Yupik-English dic-

Sequence Type	Tokenize by Character	Tokenize by Grapheme
surface form	q i k m i q	q i k mm i q
underlying form	q i k m i gh [N] [ABS] [UNPD] [SG]	q i k mm i gh [N] [ABS] [UNPD] [SG]

Table 4: Contrasts the two tokenization methods introduced in § 3.2.1 (tokenization by character and tokenization by orthographically transparent Yupik grapheme) on the surface form **qikmiq** (*dog*) and its underlying form **qikmigh**[N][ABS][UNPD][SG].

#	Possible Nouns	Possible Verbs	Both
0	1.06×10^{06}	8.80×10^{06}	9.86×10^{06}
1	1.37×10^{08}	2.04×10^{09}	2.18×10^{09}
2	2.22×10^{10}	4.19×10^{11}	4.41×10^{11}
3	4.03×10^{12}	8.31×10^{13}	8.72×10^{13}
4	7.66×10^{14}	1.63×10^{16}	1.71×10^{16}
5	1.48×10^{17}	3.18×10^{18}	3.33×10^{18}
6	2.88×10^{19}	6.21×10^{20}	6.50×10^{20}
7	5.61×10^{21}	1.21×10^{23}	1.27×10^{23}

Table 5: Number of morphotactically possible Yupik word forms formed using 0-7 derivational suffixes.

tionary and Jacobson (2001) Yupik grammar also list 3762 verb roots along with 180 intransitive and 2160 transitive verbal inflectional morphemes. If one were to naively assume that every Yupik verb can be either transitive or intransitive,[2] another 8.8 million training examples consisting of Yupik verbs could be generated.

Yupik also exhibits extensive derivational morphology. The dictionary lists 89 derivational suffixes that can each attach to a noun root and yield another noun, 58 derivational suffixes that can each attach to a noun root and yield a verb, 172 derivational suffixes that can each attach to a verb root and yield another verb, and 42 derivational suffixes that can each attach to a verb root and yield a noun. Yupik words containing up to seven derivational morphemes have been attested in the literature (de Reuse, 1994). By considering all possible Yupik nouns and verbs with up to seven derivational morphemes, well over 1.2×10^{23} inflected Yupik word forms could be generated as shown in Table 5 above. As before, many of these forms, while morphologically valid, would be syntactically or semantically illicit.

3.2 Neural Machine Translation

In this work, we made use of Marian (Junczys-Dowmunt et al., 2018), an open-source neural machine translation framework that supports bidirectional recurrent encoder-decoder models with attention (Schuster and Paliwal, 1997; Bahdanau et al., 2014). In our experiments using Marian, we trained neural networks capable of translating from input sequences of characters or graphemes (representing Yupik words) to output sequences of characters or graphemes plus inflectional tags (representing an underlying Yupik form).

3.2.1 Data

We began by preprocessing the data described in §3.1.1 by tokenizing each Yupik surface form, either by characters or by orthographically transparent, *redoubled* graphemes. An example can be seen in the transformation of the word *kaviighet* at the beginning of §3, where each character and inflectional tag were separated by a space. When tokenizing by redoubled graphemes, orthographically non-transparent graphemes were first replaced following the approach described in Schwartz and Chen (2017) which ensures there is only one way to tokenize a Yupik word form. This approach undoes an orthographic convention that shortens the spelling of words by exploiting the following facts:

- Graphemic doubling conveys voicelessness (*g* represents the voiced velar fricative while *gg* represents the *voiceless* velar fricative)
- Consecutive consonants in Yupik typically agree in voicing, with the exception of voiceless consonants that follow nasals

Yupik orthography undoubles a voiceless grapheme if it co-occurs with a second voiceless grapheme, according to the following three Undoubling Rules (Jacobson, 2001):

1. A fricative is undoubled next to a stop or one of the voiceless fricatives where doubling is not used to show voicelessness (*f*, *s*, *wh*, *h*).
2. A nasal is undoubled after a stop or one of the voiceless fricatives where doubling is not used to show voicelessness.

[2]The Yupik-English dictionary does not annotate verb roots with valence information.

3. A fricative or nasal is undoubled when it comes after a fricative where doubling is used to show voicelessness, except that if the second fricative is *ll* then the first fricative is undoubled instead.

These two tokenization methods subsequently produced two parallel corpora, whose training pairs differed as seen in Table 4 on the previous page. Nevertheless, the input data always corresponded to Yupik surface forms, and the output data corresponded to underlying forms. The parallel corpora were then randomly partitioned into a training set, a validation set, and a test set in a 0.8/0.1/0.1 ratio.

3.2.2 Initial Experiment

Using Marian, we used the data tokenized by characters and trained a shallow neural network model that implemented an attentional encoder-decoder model (Bahdanau et al., 2014) with early stopping and holdout cross validation. We used the parameters described in Sennrich et al. (2016), where the encoder and decoder consisted of one hidden layer each, of size 1024. Of the 109,395 items in the final test set, this shallow neural model achieved 100% coverage and 59.67% accuracy on the test set.

Error analysis revealed a substantial amount of underspecification and surface form ambiguity as a result of syncretism in the nominal paradigm. As exemplified in (3a) and (3b), inflectional suffixes in Yupik may share the same underlying phonological form as well as the same morphophonological rules associated with that suffix.

(3a) **ayveghet**
ayvegh- -∼$_{sf}$-w:(e)t
walrus- -**ABS**.UNPD.PL
'*walruses*'

(3b) **ayveghet**
ayvegh- -∼$_{sf}$-w:(e)t
walrus- -**REL**.UNPD.PL
'*of walruses*'

For example, any noun that is inflected for the unpossessed absolutive plural, [N][ABS][UNPD][PL], produces a word-form that is identical to the form yielded when the noun is inflected for the unpossessed relative plural, [N][REL][UNPD][PL]. The generated parallel data therefore includes the following two parallel forms, both of which have the exact same surface form. The first is the word in absolutive case:

a y v e g h e t
↓
a y v e g h [N] [ABS] [UNPD] [PL]

The second is the word in relative case:

a y v e g h e t
↓
a y v e g h [N] [REL] [UNPD] [PL]

Since these surface forms are only distinguishable through grammatical context, and our neural analyzer was not trained to consider context, it was made to guess which underlying form to return, and as suggested by the low accuracy score of 59.67%, the analyzer's guesses were often incorrect. We did not think it was proper to penalize the analyzer for wrong answers in instances of syncretism, and consequently implemented a post-processing step to account for this phenomenon.

This step was performed after the initial calculation of the neural analyzer's accuracy score, and provided an estimated or adjusted accuracy score that considered the syncretic forms equivalent. It iterated through all outputs of the neural analyzer that were initially flagged as incorrect for differing from their test set counterparts. Using the finite-state analyzer, the surface forms for each output and its corresponding test set item were then generated to verify whether or not their surface forms matched. If they matched, the neural analyzer's output was instead counted as correct (see Table 6 on the following page for examples). Assessed in this way, the shallow model achieved an adjusted accuracy score of 99.90%.

3.2.3 Data Revisited

Although the postprocessing step is sufficient to demonstrate the true performance of the neural analyzer, we attempted to resolve this ambiguity issue with a more systematic approach. In their development of a neural morphological analyzer for Arapaho verbs, Moeller et al. (2018) conflated tags that resulted in ambiguous surface forms into a single, albeit less informative, tag, such as **[3-SUBJ]**, joined from [3SG-SUBJ] and [3PL-SUBJ]. We attempted to do the same for Yupik by collapsing the tag set, but Yupik presents a somewhat more intricate ambiguity patterning. Syncretic tags can differ in their case markings alone, as in (3a) and (3b), but they can also differ across

Neural Analyzer Output	Surface	Gold Standard	Surface	
anipa[N][ABS][UNPD][PL]	anipat	anipa[N][REL][UNPD][PL]	anipat	✓
wayani[N][LOC][UNPD][PL]	wayani	wayagh[N][LOC][UNPD][PL]	wayani	✓
suflu[N][LOC][UNPD][PL]	sufluni	suflugh[N][ABS][4SGPOSS][SGPOSD]	sufluni	✓
puume[N][LOC][UNPD][DU]	pumegni	puu[N][LOC][4DUPOSS][SGPOSD]	puumegneng	✗

Table 6: An illustration of the process of the post-processing step that was implemented to resolve surface form ambiguity. If the output and its gold standard match in their surface forms, the output is then considered correct, despite the mismatch in the underlying forms.

case, possessor type, *and* number, as seen in (4a) and (4b).

(4a) **neghsameng**
neghsagh- -∼f-wmeng
seal- **-ABL_MOD.UNPD.SG**
'*seal* (as indefinite object); *from seal*'

(4b) **neghsameng**
neghsagh- -∼f-wmeng
seal- **-REL.PL.4DUPOSS**
'*their*$_2$ (reflexive) *seals*'

As a result, we could not conflate our tags in the same way Moeller et al. (2018) did. Instead, for each set of syncretic tags, one string of tags was selected to represent all of the tags in the set, such that [N][ABS][UNPD][PL] denoted both unpossessed absolutive plural *and* unpossessed relative plural. The original 273 unique strings of tags (7 cases × 13 possessor types × 3 number markers) were consequently reduced to 170 instead.

Having identified and reduced tag set ambiguity, we retrained the shallow model but only managed to achieved an unadjusted accuracy score of 95.48%. Additional error analysis revealed that some surface form ambiguity remained, but among non-syncretic tags that could not be collapsed. In other words, these tags generated identical surface forms for some nouns but not others. This is shown in (5a) – (5d):

(5a) **sufluni**
suflu- -–ni
cave- **-ABS.SG.4SGPOSS**
'*his/her* (reflexive) *own cave*'

(5b) **sufluni**
suflu- -∼f-wni
cave- **-LOC.UNPD.PL**
'*in the cave*'

(5c) **sufluni**
suflug- -–ni
chimney- **-ABS.SG.4SGPOSS**
'*his/her* (reflexive) *own chimney*'

(5d) **suflugni**
suflug- -∼f-wni
chimney- **-LOC.UNPD.PL**
'*in the chimney*'

Thus, despite the identical surface forms shown in (5a) and (5b), these same inflection tags do not result in identical surface forms for the noun root **suflug** in (5c) and (5d), since the underlying inflectional suffixes they represent are distinct: **–ni** versus **∼f-wni**. As such, these two strings of tags, [N][ABS][4SGPOSS][SGPOSD] and [N][LOC][UNPD][PL], cannot be collapsed.

Since the tag set cannot be reduced further than 170 tags, we must invoke the post-processing step introduced in §3.2.2 regardless. Moreover, since our proposed method results in some loss of information with respect to all possible underlying forms, we will have to seek an alternative method for handling syncretism. Nevertheless, after applying the post-processing step, the retrained model also achieved an adjusted accuracy score of 99.90%.

3.2.4 Additional Experiments

We trained four models on the inflected nouns dataset, experimenting with the shallow versus deep neural network architectures and the two tokenization methods: by characters and by redoubled graphemes. The shallow neural models were identical to those described in §3.2.2 and §3.2.3. The deep neural models used four hidden layers and LSTM cells, following Barone et al. (2017). As before, all models were trained to convergence and evaluated with holdout cross validation on the same test set. Results are presented in Table 7 on the next page, along with the accuracy scores before and after resolving all surface ambiguities.

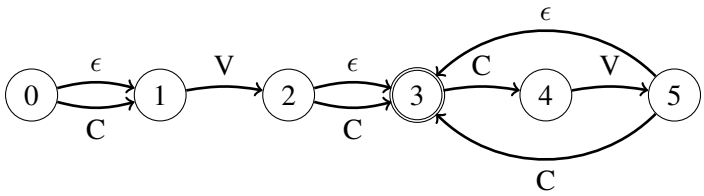

Figure 1: Finite-state diagram depicting legal word structure for nouns in Yupik. Here, **V** may refer to either a short vowel, *-i, -a, -u, -e*, or a long vowel, *-ii, -aa, -uu*, and **C** refers to any consonantal Yupik grapheme. Additionally, any noun-final **C** is restricted to be *-g*, *-gh*, *-ghw*, or *-w*.

Model	Tokenization	Accuracy	Adjusted
shallow	char	95.37	99.87
deep	char	95.07	99.95
shallow	redoubled	95.48	99.90
deep	redoubled	95.17	**99.96**

Table 7: Accuracy and adjusted accuracy scores on the generated test data from §3.2.1 (before and after resolving all surface ambiguity) for each model. The bolded percentage indicates the highest-performing model on the heldout test data.

While all models reached over 99% adjusted accuracy, the deep models outperformed their shallow counterparts, and the models trained on data tokenized by redoubled graphemes fared marginally better than those trained on data tokenized by individual characters. The latter may result from the fact that some inflections operate on full graphemes, for instance, **gh#** → **q#** during inflection for the unpossesed absolutive singular. The percentage improvement is so slight, however, that this may not be of much consequence. The deep model trained on redoubled graphemes was most accurate, peaking at 99.96%.

Despite comparable accuracy scores, there did not appear to be a discernible pattern with respect to errors among the four models.

4 Finite-State Guesser

We modified the finite-state analyzer of Chen and Schwartz (2018) by implementing a guesser module; this guesser permits the analyzer to hypothesize possible roots not present in its lexicon that nevertheless adhere to Yupik phonotactics and syllable structure. Noun roots, for example, may only end in a vowel, *-g*, *-gh*, *-ghw*, or *-w*, and follow the structural pattern given by the regular expression below:

(C) V (C) (C V (C))*

Thus, any guess made for a noun would adhere to this patterning, which the finite-state diagram in Figure 1 above captures visually. Moreover, the guesser was implemented as a backoff mechanism, such that it was only called if no other analysis could be determined. Guesses were also labeled with an additional tag, [GUESS], to distinguish them from the output returned by the finite-state analyzer itself.

5 Comparing Neural vs. Finite-State Analyzers

The final experimental condition involved comparing and quantifying the performance of the neural analyzer against its equivalent finite-state counterpart. Because the test set used in §3 was generated by the finite-state Yupik analyzer, it would be unfair to contrast the performance of the neural analyzer and the finite-state analyzer on this dataset, as the finite-state analyzer would be guaranteed to achieve 100% accuracy.

5.1 Blind Test Set

Instead, we made use of the nouns from a published corpus of Yupik interlinear glosses as an unseen test set: *Mrs. Della Waghiyi's St. Lawrence Island Yupik Texts With Grammatical Analysis by Kayo Nagai* (2001), a collection of 14 orally-narrated short stories that were later transcribed, annotated and glossed by hand. From these stories, we extracted all inflected nouns with no intervening derivational suffixes to form a modest evaluation corpus of 360 words. This was then pared down further to 349 words by removing 11 English borrowings that had been inflected for Yupik nominal cases. Manual examination of the evaluation corpus revealed several problematic items that we believe represent typos or other errors. These were removed from the final evaluation set.

for types	Coverage	Accuracy
FST (No Guesser)	85.78	78.90
FST (w/Guesser)	100	84.86
Neural	100	92.20

for tokens	Coverage	Accuracy
FST (No Guesser)	85.96	79.82
FST (w/Guesser)	100	84.50
Neural	100	91.81

Table 8: Comparison of coverage and accuracy scores on the blind test set (§5.1), contrasting the finite-state and neural analyzers. Accuracy is calculated over all types and tokens.

5.2 Blind Test Results

We analyzed the data from §5.1 using the original finite-state analyzer, the finite-state analyzer with guesser (§4), and the best-performing neural model from §3.2.4 (the deep model trained on graphemes).

Accuracy scores for the neural and finite-state analyzers, when evaluated on this refined corpus, are reported for types and tokens in Table 8 above, where accuracy was calculated over the total number of types and tokens, respectively. On the blind test set, the neural analyzer achieved coverage of 100% and accuracy of over 90%, outperforming both finite-state analyzers.

6 Discussion

Of the two analyzers which manage to achieve maximum coverage by outputting a parse for each item encountered, the neural analyzer consistently outperforms the finite-state analyzer, even when the finite-state analyzer is supplemented with a guesser. Furthermore, as illustrated in Tables 9 and 10, the neural analyzer is also more adept at generalizing.

6.1 Capacity to Posit OOV Roots

Out-of-vocabulary (OOV) roots are those roots that appear in the evaluation corpus extracted from Nagai and Waghiyi (2001) that do not appear in the lexicon of the finite-state analyzer nor in the Badten et al. (2008) Yupik-English dictionary. Of the seven unattested roots identified in the corpus, the neural analyzer returned a correct morphological parse for three of them while the finite-state analyzer only returned two (see Table 9).

Unattested Root	FST	NN
aghnasinghagh	–	–
aghveghniigh	–	✓
akughvigagh	✓	✓
qikmiraagh	–	–
sakara	✓	–
sanaghte	–	–
tangiqagh	–	✓

Table 9: Comparison of finite-state and neural analyzer's performances on unattested roots, which are unaccounted for in both the lexicon of the finite-state analyzer and the Badten et al. (2008) Yupik-English dictionary. A checkmark indicates that the correct morphological analysis was returned by that analyzer.

6.2 Capacity to Handle Spelling Variation

The neural analyzer performed even better with spelling variation in Yupik roots (see Table 10). Of the three spelling variants identified in the corpus, all of them differed from their attested forms with respect to a single vowel, *-i-* versus *-ii-*. The neural analyzer returned the correct morphological parse for all spelling variants, while the finite-state analyzer supplemented with the guesser only succeeded with one, **melqighagh**. Moreover, the neural analyzer even managed to guess at the correct underlying root in spite of a typo in one of the surface forms (*uk**u**sumun* rather than *uksumun*).

Root Variant	FST	NN
melq**i**ghagh	✓	✓
piites**ii**ghagh	–	✓
uqf**ii**lleghagh	–	✓
*uk**u**sumun	–	✓

Table 10: Comparison of finite-state and neural analyzer's performances on root variants, which are spelled differently from their attested counterparts in the lexicon of the finite-state analyzer and Badten et al. (2008). A checkmark implies that the analyzer returned the correct morphological analysis while the asterisk * denotes an item with a typo.

6.3 Implications for Linguistic Fieldwork

The higher-quality performance of the neural analyzer has immediate implications for future computational endeavors and field linguists working in Yupik language documentation and longer-range implications for field linguists in general. With respect to fieldwork, a better-performing analyzer with greater and more precise coverage equates

to better real-time processing of data for field linguists performing morphological analysis and interlinear glossing.

The neural analyzer is also immune to overgeneration, since it relies on a machine translation framework that returns the "best" translation for every input sequence; in our case, this equates to one morphological analysis for each surface form. This contrasts with the finite-state analyzer variants that may return hundreds or thousands of analyses for a single surface form if the finite-state network permits. For instance, it was found that within the text corpus of the Jacobson (2001) reference grammar alone (see Table 1), the Yupik finite-state analyzer (without guesser) generated over 100 analyses for each of 16 word types. The word with the greatest number of analyses, **laalighfiknaqaqa**, received 6823 analyses followed by **laalighfikiikut** with 1074 (Chen and Schwartz, 2018). In this way, to have developed a neural analyzer that returns one morphological analysis per surface form is valuable to field linguists as it does not require them to sift through an indiscriminate number of possibilities.

6.4 Application to Other Languages

Aside from our procedure to tokenize Yupik words into sequences of fully transparent graphemes, exceedingly little language-specific preprocessing was performed. Our general procedure consists of creating a parallel corpus of surface and underlying forms by iterating through possible underlying forms and using a morphological finite-state transducer to generate the corresponding surface forms. We believe that this procedure of bootstrapping a learned morphological analyzer through the lens of machine translation should be generally applicable to other languages (especially, but certainly not exclusively, those of similar typology).

6.5 Added Value of FST Analyzers

Finally, the methodology employed here, in which a neural analyzer is trained with data generated from an existing finite-state implementation, is inherently valuable. Though the development of finite-state morphological analyzers demands considerable effort, the fact that their output may be leveraged in the development of better-performing systems is especially practical for under-resourced languages such as Yupik, where any form of training data is scarce. Thus, finite-state analyzers may serve a twofold purpose: that of morphological analysis, as they were intended to be used, but also for the generation of training data to train neural systems.

7 Conclusion

Morphological analysis is a critical enabling technology for polysynthetic languages such as St. Lawrence Island Yupik. In this work we have shown that the task of learning a robust high-accuracy morphological analyzer can be bootstrapped from an existing finite-state analyzer. Specifically, we have shown how this can be done by framing the problem as a machine translation task. We have successfully trained a neural morphological analyzer for derivationally unaffixed nouns in St. Lawrence Island Yupik, and compared its performance with that of its existing finite-state equivalent with respect to accuracy.

This work represents a case where the student truly learns to outperform its teacher. The neural analyzer produces analyses for all Yupik word types it is presented with, a feat that the original finite-state system fails to achieve. At the same time, the neural analyzer achieves higher accuracy than either the original finite-state analyzer or a variant FST augmented with a guesser. The neural analyzer is capable of correctly positing roots for out-of-vocabulary words. Finally, the neural analyzer is capable of correctly handling variation in spelling.

In future work, we plan to explore more thorough methods for handling ambiguous surface forms. We also plan to correct the minor FST error identified in §3.1.2. Most importantly, the training dataset will be extended to include items beyond inflected nouns with no intervening derivational suffixes. Specifically, we intend to increase the training set to include verbs, particles, and demonstratives in addition to nouns, as well as words that include derivational suffixes.

References

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

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. 1994. *Akiingqwaghneghet / Echoes*. Bering Strait School District, Unalakleet, Alaska.

Anders Apassingok, (Iyaaka), Jessie Uglowook, (Ayuqliq), Lorena Koonooka, (Inyiyngaawen), and Edward Tennant, (Tengutkalek), editors. 1995. *Suluwet / Whisperings*. Bering Strait School District, Unalakleet, Alaska.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. 1985. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 1: Gambell. Bering Strait School District, Unalakleet, Alaska.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. 1987. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 2: Savoonga. Bering Strait School District, Unalakleet, Alaska.

Anders Apassingok, (Iyaaka), Willis Walunga, (Kepelgu), and Edward Tennant, (Tengutkalek), editors. 1989. *Sivuqam Nangaghnegha — Siivanllemta Ungipaqellghat / Lore of St. Lawrence Island — Echoes of our Eskimo Elders*, volume 3: Southwest Cape. Bering Strait School District, Unalakleet, Alaska.

Linda Womkon Badten, (Aghnaghaghpik), Vera Oovi Kaneshiro, (Uqiitlek), Marie Oovi, (Uvegtu), and Christopher Koonooka, (Petuwaq). 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*.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. *arXiv preprint arXiv:1707.07631*.

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 (LREC'18)*, 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*, 2nd edition. Alaska Native Language Center, University of Alaska Fairbanks, Fairbanks, Alaska.

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.

Christopher (Petuwaq) Koonooka. 2003. *Ungipaghaghlanga: Let Me Tell You A Story*. Alaska Native Language Center.

G.A. Menovshchikov. 1988. *Materialiy i issledovaniia po iaziku i fol'kloru chaplinskikh eskimosov.* Nauka, Leningrad. Traditional Yupik stories, with stress and length indicated; includes Russian translation.

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.

Kayo Nagai and Della Waghiyi. 2001. *Mrs. Della Waghiyi's St. Lawrence Island Yupik texts with grammatical analysis*, volume A2-006 of *Endangered Languages of the Pacific Rim Publications Series*. ELPR, Osaka.

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 and Emily Chen. 2017. Liinnaqumalghiit: A web-based tool for addressing orthographic transparency in St. Lawrence Island/Central Siberian Yupik. *Language Documentation and Conservation*, 11:275–288.

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