A Morphological Analyzer for St. Lawrence Island / Central Siberian Yupik

Emily Chen, Lane Schwartz
Department of Linguistics
University of Illinois Urbana-Champaign, Champaign, Illinois
{echen41, lanes}@illinois.edu

Abstract

St. Lawrence Island / Central Siberian Yupik is an endangered language, indigenous to St. Lawrence Island in Alaska and the Chukotka Peninsula of Russia, that exhibits pervasive agglutinative and polysynthetic properties. This paper discusses an implementation of a finite-state morphological analyzer for Yupik that was developed in accordance with the grammatical standards and phenomena documented in Steven A. Jacobson's 2001 reference grammar for Yupik. The analyzer was written in *foma*, an open source framework for constructing finite-state grammars of morphology. The approach presented here cyclically interweaves morphology and phonology to account for the language's intricate morphophonological system, an approach that may be applicable to languages of matching typology. The morphological analyzer has been designed to serve as foundational resource that will eventually underpin a suite of computational tools for Yupik to assist in the process of linguistic documentation and revitalization.

Keywords: computational morphology, finite-state grammar, morphological analyzer

1. Introduction

We introduce in this paper an implementation of a morphological analyzer for St. Lawrence Island / Central Siberian Yupik (ISO 639-3: ess), an agglutinative, polysynthetic language of the Inuit-Yupik language family. [1] This analyzer is implemented in *foma* (Hulden, 2009b), and represents a faithful adaptation of the grammatical and morphophonological rules documented in *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language* (Jacobson, 2001). We intend for the morphological analyzer to be a foundational resource that underpins a suite of computational tools to be shared with the Yupik community for purposes of language preservation and revitalization.

2. Language Description

Yupik is the westernmost variety of the Inuit-Yupik language family (Krauss et al., 2010). Most of the 2400–2500 Yupik people reside in two villages on St. Lawrence Island, Alaska and two villages on the Chukotka Peninsula of far eastern Russia (Krupnik and Chlenov, 2013). Out of that population, fewer than 1000 are estimated to be L1 Yupik speakers (Koonooka, 2005; Schwalbe, 2017).

The Yupik phonemic inventory comprises 31 consonants and 4 vowels, /ə/, /i/, /a/, /u/ which can be lengthened (with the exception of /ə/) for a total of seven vocalic phonemes. Of the 31 consonants, there are 8 pairs of continuants and 4 pairs of nasals where each pair differs only in voicing. In the standard Latin-based orthography, this difference is marked in 5 of the 8 continuant pairs and in all of the nasal pairs through *graphemic doubling*, that is, *l* and *ll*, *r* and *rr*, *g* and *gg*, *gh* and *ghh*, *ghw* and *ghhw*, *m* and *mm*, *n* and *nn*, *ng* and *ngng*, *ngw* and *ngngw*.

2.1. Morphology

Like all languages in the Inuit-Yupik family, the morphology of Yupik is remarkably generative. Yupik nouns and verbs are principally responsible for the most morphologically complex words in the language, and permit up to seven derivational morphemes or *postbases* as they are referred to in the literature (de Reuse, 1994, p.53). Noun and verb roots are considered *bases*, although the term *base* is used more generally to apply to any uninflected form that is available for further affixation (de Reuse, 1994, p.24). That is to say, a *base* + *postbase* unit may also be referred to as a *base*. There is only one attested prefix in Yupik, which can be applied only to demonstratives; all other affixation is suffixing (Jacobson, 2001, p.109). Thus, the underlying structure of most Yupik words is *(noun/verb) base + zero or more derivational postbases + inflectional postbase* (+ *optional enclitic*), where enclitics are associated with nouns only and simply affix in word-final position.

As far as previous literature has shown, derivational postbases in Yupik can only be one of four types, and may or may not constitute a closed class:

1. *Nominalizing* postbases affix to verbs and yield nouns.
2. *Noun-elaborating* postbases affix to nouns and yield nouns.
3. *Verbalizing* postbases affix to nouns and yield verbs.
4. *Verb-elaborating* postbases affix to verbs and yield verbs.

Nouns subsequently inflect for person, number, and possession based on the case of the noun base, while verbs inflect for person and number based on the mood of the verb base. Person and number are expressed within a single morpheme, suggesting that Yupik also possesses some fusional properties. Other root bases that can be inflected include demonstratives, numerals, and personal pronouns, while Yupik "adjectives" manifest as verb bases, such as **kavite-** (*to be red*).

2.2. Morphophonology

While some Yupik postbases directly affix to bases, such as the verb-elaborating postbase **+sug** (*to want one to V*) which yields **kavitesug-** (*to want one to be red*), most postbases trigger a series of morphophonological changes at the

[1] The term *Yupik* will be used henceforth to refer to the St. Lawrence Island / Central Siberian Yupik variety.

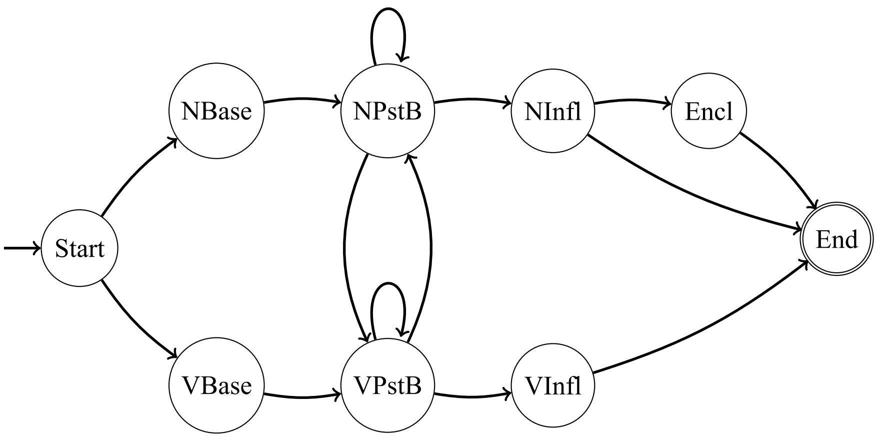

Figure 1: Finite-state network that illustrates the general design of the Yupik analyzer with the exception of some root bases and continuation classes, such as demonstratives and numerals. Of the ones shown, `NBase` and `VBase` refer to the noun base and verb base lexicons, while `NPstB` and `VPstB` refer to the respective postbase lexicons. The transition arcs between the postbase lexicons account for nominalizing and verbalizing postbases, while the `NInfl` and `VInfl` continuation classes handle case and mood inflection, and `Encl` handles enclitics.

immediate left-adjacent base-postbase boundary. In (Jacobson, 2001), each morphophonological process is systematically documented and assigned a unique symbol, such as + which designates straightforward affixation of postbase to base. While the morphonological processes called upon by each postbase are not predictable, the symbols representing each process are somewhat lexicalized, and hint at the process being represented (see Table 1). In all examples we use the standard listing order and notation used in the Yupik grammar (Jacobson, 2001) and the Yupik dictionary (Badten et al., 2008) as well as the Leipzig glossing conventions. For the most part, the ordering of the morphophonological symbols listed with each postbase reflects the order in which the morphophonological processes apply, as in **@$\sim_f$–ragkiigh** (*to V quickly*). Modification of base-final *te*, which is represented by the symbol **@**, occurs before base-final *e* is dropped via $\sim_f$. Otherwise, the ordering of the symbols is arbitrary, since the morphophonological processes being represented are mutually exclusive, as in $\sim_f$ and **–**, where the latter drops base-final consonants. As a base cannot end in a consonant and an *-e*, the only necessary symbol ordering in the postbase **@$\sim_f$–raghkiigh** is between **@** and $\sim_f$. Any allomorphy in the postbase however, is always handled first.

Following these conventions, the subsequent example illustrates the full derivation of the Yupik word **aghnaaguq**.

(1) aghnaaguq

aghnagh-	-∼:(ng)u-	-$\sim_f$(g/t)u-	-q
woman-	-to.be.N-	-INTR.IND-	-3SG

'*She is a woman*'

The verbalizing postbase **∼:(ng)u**, with its morphophonological processes underlined, affixes first:

1. **(ng)** If the base ends in a vowel, affix *ng*
2. ∼ If *e* appears in base-final or penultimate (semi-final) position, drop *e*
3. **:** If *gh* appears between two vowels that can be lengthened, drop *gh*

Following this first iteration, the resulting intermediate form is **aghnau-** (*to be a woman*). Yupik phonology forbids unlike vowel clusters however, so *u* assimilates to *a* in a process known as *vowel dominance* to yield **aghnaa-**.
The affixation of the postbase **$\sim_f$(g/t)u** marks the valence and verb mood as intransitive indicative:

1. **(g/t)** If the base ends in a vowel or a consonant, affix allomorphs *g* or *t* respectively
2. $\sim_f$ If base ends in final *e*, drop *e*

Finally the person/number marker **q** completes the surface form. Thus, from the string *aghnagh-∼:(ng)u-$\sim_f$(g/t)u-q*, we derive **aghnaaguq**.

2.3. Derivation of a More Intricate Word

Here, we derive a Yupik word of greater complexity, **pagunghalighnaqaqa** (*I am going to put crowberries in…*), that consists of two derivational postbases, and four unique morphophonological processes contained within those postbases. The underlying gloss and source sentence are given in Example 2, while a full listing of the standard morphophonological symbols is also provided in Table 1 as a reference.
The verbalizing postbase **–ligh**, with its single morphophonological process underlined, affixes first to the base **pagunghagh***.

1. **–** If the base ends in a consonant, including strong *gh* (see Table 1), drop the consonant

Following this iteration, the first intermediate form is **pagunghaligh-** (*to put crowberries in*).
Affixation of the verb-elaborating postbase **@$\sim_f$naqe** is next:

(2) Naagpek sagnegha

naagpek	sagnegha
naa––gpek-	-saghnegh∼:(ng)a-
mother-2SG.POSS	bowl-3SG.POSS

pagunghalighnaqaqa				
pagunghagh*-	-–ligh-	-@$\sim_f$naqe-	-∼(g)a-	-qa
crowberry-	-to.put.N.in-	-to.be.going.to.V-	-TRNS.IND-	-1SG.3PL

‘*I am going to put crowberries in…*’ (Jacobson, 2001)

Symbol	Description
∼	Drops *e* in penultimate (semi-final) position or *e* in base-final position and *hops* it *e-Hopping* is the process by a which a full vowel in the first syllable of the base is 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 would result 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 explicitly marked without an *. Strong *gh* is denoted as *gh**
:	Drops uvulars that appear between single vowels
–	Drops final consonants
––	Drops final consonants and preceding vowel
@	Indicates some degree of modification to base-final *te*, the degree of which is unpredictable and dependent on the postbase
+	Adds ending as presented This symbol is excluded from the Yupik analyzer and is implicitly assumed when no other morphophonological symbols are present.

Table 1: List of all documented morphophonological processes in Yupik and their lexicalized symbols

1. **@** For this postbase, if the base ends in *te*, drop *te*
2. **$\sim_f$** If the base ends in final *e*, drop *e*

The second intermediate form is then **pagunghalighnaqe-** (*to be going to put crowberries in*).

The affixation of the inflectional postbase **∼(g)a** marks the valence and verb mood as transitive indicative:

1. **(g)** If the base ends in a double vowel, affix *g*
2. ∼ If *e* appears in base-final or penultimate (semi-final) position, drop *e*

The last intermediate form, **pagunghalighnaqa-**, is then followed by affixation of the person/number marker **-qa**, which completes the derivation.

3. Finite State Morphology: *foma* and *lexc*

A finite state transducer is the ideal mechanism for computationally modeling morphology, since it maps a bidirectional relation between two sets of strings, that is, an underlying form and a surface form (Beesley and Karttunen, 2003).

We implemented our morphological analyzer in *foma* (Hulden, 2009b) which is responsible for the composition of string-transforming rules to derive the surface string from the underlying string and vice versa. We encode the lexicon in the *lexc* format used by *foma*. Each lexical item is associated with a *continuation class*, and is encoded with an underlying form and an intermediary form that passes through the transducer in *foma* to generate the surface form.

Also included within *lexc* are the definitions of multicharacter symbols, typically glossing abbreviations such as [N] and [V], as well as any multi-graphemic phonemes (§ 2.).

4. Implementation

The Yupik analyzer is strictly implemented within the *lexc* and *foma* languages, although lexical transducers for sister languages to Yupik such as Iñupiaq have incorporated other resources such as XML databases (Bills et al., 2010), and an analyzer for Inuktitut developed by the Institute for Information Technology of the National Research Council of Canada was implemented in Java (Institute for Information Technology, 2012). Limiting the programming of the analyzer to *foma* was sufficient for our purposes however, as there are a number of APIs to bridge the completed transducer with external utilities such as a spell-checker (Hulden, 2009a).

4.1. *lexc* File

The (Jacobson, 2001) reference grammar contains approximately 600 root bases and 80 derivational morphemes, in addition to the extensive inflectional morphology for noun case and verb mood. From this, the *lexc* file was crafted by hand to ensure that each lexicon was followed by the proper continuation class to generate a comprehensive set of permissible underlying strings (Figure 1).

The underlying forms and intermediate forms involved in the derivation of **aghnaaguq** are displayed in Figure 2. The noun base **aghnagh** is selected from the `NounBase` root lexicon, and proceeds to the `NounPostbase` continuation class, where it concatenates with the underlying form of the verbalizing postbase, **∼:(ng)u[N→V]**. The string derived thus far then continues to the `VInfl` continuation class, and is concatenated with the underlying gloss of the inflectional

```
LEXICON NounBase
aghnagh NounPostbase;                !woman

 LEXICON NounPostbase
–ghhagh[N→N]:–ghhagh NounPostbase; !dear N
∼%:(ng)u[N→V]:∼%:(ng)u VerbInfl;  !to be N

LEXICON VerbInfl
[V][Ind][3Sg]:∼f(g/t)uq #;
```

Figure 2: Sample *lexc* file that traces the underlying strings of '*aghnaaguq*' and '*aghnaghaaguq*' (see Examples 1 and 3). Exclamation points mark the beginning of a comment, and the percent sign is an escape character.

ending, [V][INTR][IND][3SG]. Eventually, the underlying forms of the postbase and inflectional ending, which are included for clarity and readability, are rewritten as the intermediate forms, **∼:(ng)u** and $\sim_f$**(g/t)uq** respectively, for processing in *foma*.

The noun-elaborating postbase **–ghhagh** is likewise included in the `NounPostbase` lexicon, where the morphophonological symbol – drops all base-final consonants. Its continuation class is the `NounPostbase` lexicon itself, and this form of self-reference permits the recursive attachment of derivational postbases to yield strings such as the one show below:

(3) aghnaghhaaguq

aghnagh-	–ghhagh-	-∼:(ng)u-	-$\sim_f$(g/t)u-	-q
woman-	-dear.N-	-to.be.N-	-INTR.IND-	-3SG

'*She is a dear woman*'

4.1.1. Flag Diacritics

The natural valency of a Yupik verb dictates the form of the inflectional ending, since the ending marking intransitivity differs from the ending marking transitivity with respect to the morpheme and morphophonological processes. For instance, the indicative mood postbase for intransitives is $\sim_f$**(g/t)u**, while the analogous ending for transitives is **∼(g)a**.

Foma includes a feature called *flag diacritics* to handle long-distance dependencies in words, effectively constraining the morphemes that may co-occur. Each flag diacritic has the form @FLAGTYPE.FEATURE.VALUE@, where *feature* and *value* refer to arbitrary strings set by the programmer (Hulden, 2011). A flagtype value of P indicates that the *feature* should be set to *value*, while a flagtype value of R requires the *feature* to already be set to *value*. We use the following flag diacritics:

1. @P.VALENCE.INTR@
2. @P.VALENCE.TRNS@
3. @R.VALENCE.INTR@
4. @R.VALENCE.INTR@

Strings that contain morphemes with mismatched diacritics are discarded, as demonstrated in the expanded *lexc* file presented in Fig. 3. For instance, transitive verb **ungipaate** (*to tell*) has its VALENCE feature set to TRNS via its continuation class, `VerbTrns`. It may optionally receive a verbal postbase, or directly proceed to inflection where its VALENCE feature is checked in the `VerbInfl` continuation class. A mismatch in flag diacritics then prevents the concatenation of **ungipaate** with the intransitive inflectional ending. Similar circumstances arise for intransitive verb **umughqaa** (*to have sleep paralysis*), while ambitransitive verb **nagate** (*to listen*) recognizes both endings.

```
LEXICON NounBase
aghnagh NounPostbase;             !woman

LEXICON VerbBase
nagate VerbTrns;    !to listen
ungipaate VerbTrns; !to tell
nagate VerbIntr;    !to listen
umughqaa VerbIntr;  !to have sleep paralysis

LEXICON NounPostbase
–ghhagh[N→N]:–ghhagh NounPostbase;  !dear N
∼%:(ng)u[N→V]:∼%:(ng)u VerbIntr;    !to be
N

LEXICON VerbTrns
@P.VALENCE.TRNS@ VerbPostbase;

LEXICON VerbIntr
@P.VALENCE.INTR@ VerbPostbase;

LEXICON VerbPostbase
0:0 VerbInfl;
@lleqe[V→V]:@lleqe VerbPostbase; !will V

LEXICON VerbInfl
@R.VALENCE.TRANS@ VerbInflTrns;
@R.VALENCE.INTR@  VerbInflIntr;

LEXICON VerbInflTrns
[V][Ind][3Sg][3Sg]:∼(g)aa #;

LEXICON VerbInflIntr
[V][Ind][3Sg]:∼f(g/t)uq #;
```

Figure 3: Sample *lexc* file that is expanded to include the flag diacritic continuation classes to oversee verb valence constraints.

4.2. *foma* File

The morphophonological processes responsible for transforming strings are individually implemented in the *foma* file as rules that trigger the pertinent transformation under specific environmental conditions. A subset of these rules are shown in Figure 4. Rules take the form $A \rightarrow B \mid\mid \Gamma\_\Delta$, where A is rewritten as B in the context of Γ and Δ. In the Yupik analyzer, A typically refers to the substring that is rewritten as B in the morphophonological context determined by Γ and Δ, where Δ may refer to the morphophonological symbol. As such, the *foma* file consists of all contextual rewrite rules defined according to the morphophonological processes they represent, concluding with a single rules cascade that composes the rewrite rules together in the

```
read lexc xample.lexc
⋮
define ResolveAllomorphy
  "(ng)" -> ng || V _ .o.
  "(ng)" -> 0 || C _ .o.
  "(g/t)" -> g || V _ .o.
  "(g/t) -> t || C _ .o.
  "(ng)" -> 0 .o.
  "(g/t)" -> 0;
⋮
define UvularDropping
  "gh" -> 0 || [V - e] _ ":" [V - e] .o.
  ":" -> 0;
⋮
define Grammar [
  Lexicon .o.
  ResolveAllomorphy .o.
  SemiAndFinalE .o.
  UvularDropping .o.
  FinalE .o.
  VowelDominance ];
```

Figure 4: Sample *foma* file that contains implementation of some morphophonological rules, composed together with the '.o.' operator to form a finite-state grammar.

order that the morphophonological processes occur. In accordance with the discussion in § 2.2., the contextual rewrite rule modifying base-final *te* appears earlier in the cascade relative to the rule that drops base-final *e*.

The full finite-state grammar represents the composition of the *lexc* lexicon with these contextual rewrite rules.

Yupik morphophonology requires that these rewrite rules be completely applied at each successive morpheme boundary in sequence. As such, the grammar shown in Figure 4 is insufficient to correctly process most words. When processing the underlying form from Example 1, the grammar in Figure 4 incorrectly yields **aghnaauq** as the surface form, instead of **aghnaaguq**. In this example, the morphophonological symbols *(ng)* and *(g/t)* in the underlying string *aghnagh-∼:(ng)u-∼f(g/t)u-q* simultaneously delete to no effect, since neither appear in the contexts specified in the transducer. The premature deletion of *(g/t)* falsely yields **aghnaauq**. Although the rules in Figure 4 are composed in the correct order, the fact that the rules are applied to all morpheme boundaries simultaneously results in an incorrect derivation.

To remedy this, we configured the transducer to resolve the morphophonological processes of each successive morpheme boundary in its entirety before considering the morphophonological processes of the next boundary. This was accomplished by having the original single cascade of morphophonological processes iterate eight times in anticipation of seven potential derivational postbases (§ 2.1.) and subsequent inflection. Each character string used in *foma* was further categorized as either an alphabetic character (*Alphabet*) or a morphophonological symbol (*MorphPhonSymbol*), and a morpheme boundary marker ^ was introduced at every juncture of an alphabetic character and a morphophonological symbol. The required context for each rule was modified such that the rule applies only at the leftmost unprocessed morpheme boundary. At the end of each iteration, the leftmost morpheme boundary marker, and the associated morphophonological symbols, are permitted to delete, thus setting up the requisite context for the next iteration. Figure 5 illustrates these changes.

```
read lexc xample.lexc

define Alph [ "*" | a |…| y ];
define MPSymbols [ (g/t) | (ng) |":"|…];
define MBndry "^";
define WBndry [ .#. ];

define InsertMBndry
  [..] -> MBndry || Alphabet _ MPSymbols;
define CleanupMBndry
  MBndry -> 0 || WBndry Alph+ _;
⋮
define ResolveAllomorphy
  "(ng)" -> "ng" || V MBndry _ .o.
  "(ng)" -> 0 || C MBndry _ .o.
  "(g/t)" -> g || V MBndry _ .o.
  "(g/t) -> t || C MBndry _ .o.
  "(ng)" -> 0,
  "(g/t)" -> 0 || WBndry Alph+ MBndry _;
⋮
define UvularDropping
  "gh" -> 0 || [V - e] _ MBndry MPSymbols*
":" [V - e] .o.
  ":" -> 0 || WBndry Alph+ MBndry _ ;
⋮
define Grammar [
  Lexicon .o.
  InsertMBndry .o.
  !! ITERATION 1 !!
  ResolveAllomorphy .o.
  SemiAndFinalE .o.
  UvularDropping .o.
  FinalE .o.
  VowelDominance .o.
  CleanupMBndry .o.
    ⋮
  !! ITERATION 8 !!
  ResolveAllomorphy .o.
  SemiAndFinalE .o.
  UvularDropping .o.
  FinalE .o.
  VowelDominance .o.
  CleanupMBndry ];
```

Figure 5: Sample *foma* file that correctly derives the surface string '*aghnaaguq*', where '+' regex operator denotes one or more of the preceding string.

The derivation of the string **aghnaaguq** now models the it-

(4) Piyukuvek qergesek qelpeghtikek

piyukuvek	qergesek	qelpeghtikek
piyug[V][Intr][Cond][2Sg]	**qergese[N][Abs][Unpd][Du]**	**qelpeghte[V][Trns][Opt][PRS][2Sg][3Du]**
piyug-@$\sim_{sf}$–(g)k(u)vek	qergese-$\sim_{sf}$-w:(e)-k	qelpeghte-$\sim_{f}$(i)kek
person.walking.in.distance-INTR.COND.2SG	window-UNPD.ABS-2DU	to.open-TRNS.PRS.OPT.2SG.3DU

‘*Open the window(s) if you (take a) walk*’ (Jacobson, 2001)

(5) Tukuqa neghsameng gaaghaquq

tukuqa	neghsameng	gaaghaquq
tukugh[N][Abs][1SgPoss][SgPosd]	**neghsagh[N][Abl_Mod][Unpd][Sg]**	**gaagh$\sim$(g_1)aqe[V→V][V][Intr][Ind][3Sg]**
tukugh-$\sim$–ke	neghsagh-$\sim_{f}$-wmeng	gaagh-$\sim$(g_1)aqe-$\sim_{f}$(g/t)u-q
host-1SG.POSS	seal-UNPD.ABL_MOD.SG	to.cook-to.be.Ving-INTR.IND-3SG

‘*My host is cooking seal*’ (Jacobson, 2001)

(6) Mangteghaghllangllaghyugtukut
mangteghagh–-ghllag[N→N]–-ngllagh[N→V]-@$_1$ $\sim_{f}$yug[V→V][V][Intr][Ind][1Pl]
mangteghagh–-ghllag–-ngllagh-@$\sim_{f}$yug-$\sim_{f}$(g/t)u-kut
house-huge.N-to.build.N-to.want.to.V-INTR.IND-1PL
‘*We want to build a huge house*’ (Jacobson, 2001)

	Precision	Recall	F-Measure
Types	97.82	97.61	97.46
Tokens	98.20	97.11	97.90

Table 2: Reported precision, recall, and f-measure values of the Yupik analyzer when evaluated against the end-of-chapter translation exercises of the reference grammar.

erative process first described in § 2.2.. From the underlying string *aghnagh^$\sim$:(ng)u^$\sim_{f}$(g/t)uq*, the first postbase applies in its entirety which allows the leftmost morpheme boundary marker to delete, but retains the morphophonological symbol *(g/t)* until application of the second and final postbase.

5. Evaluation

While the Jacobson (2001) reference grammar was instrumental to the implementation of the analyzer, the grammar nonethless was somewhat lexically-impoverished, listing no more than 600 noun and verb bases and 80 postbases of the approximately 8000 nouns and verbs and 600 postbases documented in the Yupik dictionary. In order to more precisely evaluate the efficacy of the analyzer, the remaining lexical entries were added to the *lexc* file via a semi-automated process that organized the dictionary entries by type, that is, as either a noun, verb, particle, or one of the four derivational postbases. Placement into one of these classes was determined by exploiting patterns in the dictionary definitions, for instance, the definition of a nominalizing postbase typically contained the phrase “one who…” or “one that…”, while the definition of a verb-elaborating postbase contained “to…V”. In all, some 4000 noun bases, 4000 verb bases, 600 postbases, and 500 particles were newly integrated into the analyzer’s lexicon. As a result, when evaluated against the end-of-chapter translation exercises offered in the reference grammar, the morphological analyzer performed reasonably well, successfully parsing several hundreds of words of varying morphological complexity (see Examples 4–6).

Numerically, the end-of-chapter exercises consisted of 281

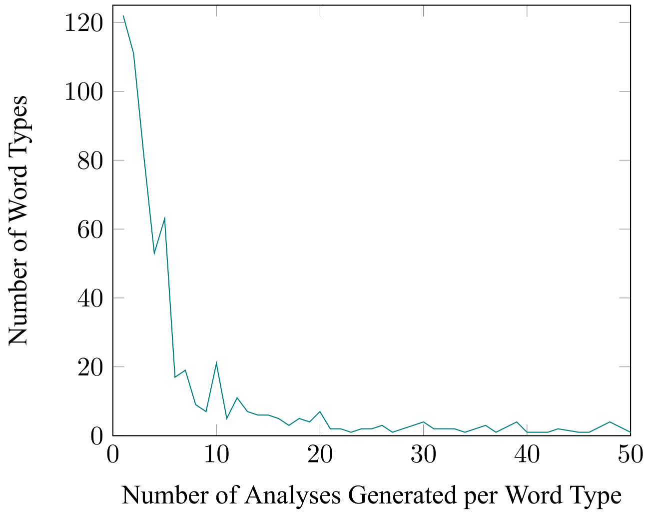

Figure 6: Of the Yupik word types in the Jacobson (2001) end-of-chapter exercises for which the analyzer returns a result, the analyzer returns five or fewer analyses for 67%. There are 30 Yupik word types from these exercises for which the analyzer returns more than 50 analyses.

Yupik sentences to be translated by the reader, summing to 796 individual tokens, of which 658 were unique. The analyzer could not derive parses for 19 tokens, while gold standard translations provided by a native Yupik speaker suggested that all of the analyses returned for 14 of the 658 unique tokens were incorrect.

Precision, recall, and f-measure values were then calculated for types versus tokens and are reported in Table 2.

While these values attest to the coverage of the analyzer, its efficiency and accessibility was further assessed by examining its output for all parsed words.

In particular, the average number of analyses generated per word was 20, while the median was only five, a discrepancy that suggested some words in the end-of-chapter exercises were generating an unusually high number of analyses, thus distorting the average. To qualify these observances, a script was implemented to identify outliers, and

(7) Laalighfikiikut
***laaligh—–i[N→V]-@$_1$ ∼:(u)n[V→N]–-ghte[N→V]-@$_1$ ∼$_f$vik[V→N]—–i[N→V][V][Intr][Ind][1Pl]**

51	tukfighinaluni
54	inimakanga
54	qikmilguyugtunga
59	lliiki
62	seghleghunii
63	ungipaatinga
67	laalightaqngavnga
72	qavaghniinga
76	tagitiki
81	mangteghaanituq
86	atightughaqsin
92	aghveliighsiin
92	naliita
98	kiyaghhneghinniluki
125	aakaqa
136	neghyaqunaan
137	pagunghalighnaqaqa
141	ungipaataanga
143	guunnaqaqa
185	atuqnaqegkeni
200	esghaataqukut
215	liilleqii
270	nallukaqa
315	atuqnaqaa
364	akitutaqukung
383	alikaqa
414	aghvightesugiinkut
461	llinaqaqa
1074	laalighfikiikut
6823	laalighfiknaqaqa

Table 3: Yupik word types from the Jacobson (2001) end-of-chapter exercises for which the analyzer returns more than 50 analyses, along with the number of analyses returned for each such word type.

found that the word with the greatest number of analyses **laalighfiknaqaqa** had *6823* analyses followed by **laalighfikiikut** with 1074. Although these values suggest that the quality of output of the morphological analyzer was at times substandard, instances of acute overgeneration were relatively rare (see Figure 6 and Table 3). Of the 658 unique tokens in the corpus, 136 tokens or approximately 20.64% generated more than ten analyses, while 77 tokens or approximately 11.68% generated more than 20 analyses. Concerning more severe cases, only 16 tokens or approximately 2.43% generated more than 100 analyses.

6. Ongoing Work

As a result of these evaluations, the most pressing issue at present concerns curtailing the instances of severe overgeneration, and adapting the analyzer to generate output that is minimal but correct, and amenable to incorporation into a Yupik language spell-checker and pedagogical materials for students. Fortunately, as demonstrated by the **laalighfiknaqaqa** and **laalighfikiikut** case examples, there seems to exist a pattern among those words that are most susceptible to overgeneration, and identifying these patterns may assist in paring down the number of analyses generated per word. For instance, one proposed nonsense analysis of the word **laalighfikiikut** is given in Example 7, where the verbalizing postbase **—i[N→V]** (*to make N*) materializes twice, resulting in semantic absurdity. Resolving this could be as simple as programming a `Filter` function that filters out any undesired permutations of strings, although the feasibility of such a solution would require a better understanding of the scope to which this overgeneration occurs.

It is likewise critical that the analyzer eventually be evaluated against texts other than the reference grammar, to ensure completeness of the dictionary, and of the morphophonological rules presented in the grammar. We have already identified several gaps in the documentation concerning certain Yupik linguistic phenomena, including aspects of the demonstrative and numeral systems. In particular, the reference grammar presents demonstrative inflection as a series of eleven “paradigmatic sets”, and within each set, demonstratives take on an additional six base forms that dictate the inflected form (Jacobson, 2001, p.110). It is unclear however, how these sets and base forms were determined, and to what degree they differ from one another with respect to inflection. Concerning Yupik numerals, the reference grammar says little beyond the fact that they may serve as appositives to other nouns, incorporate into verbs via verbalizing postbases, and may also inflect for the specific *ablative-modalis* case (Jacobson, 2001, p.112), raising the issue of whether there are limitations to the suffixing patterns of numerals. Since understanding these systems may require a considerable sum of fieldwork and elicitation, these systems are, at the moment, simply hard-coded into the *lexc* file. A more elegant and desired solution would designate each of these grammatical elements as individual `Root` lexicons with the relevant continuation classes that reflect the element’s inflection patterns.

Lastly, it is our intention to eventually extend this morphological analyzer to parse Yupik written in the Cyrillic script as well. We imagine that the morphophonological rules cascade should remain relatively the same, although the graphotactic constraints differ, for instance, the Cyrillic Yupik orthography differentiates rounded consonants from their unrounded counterparts by means of the Cyrillic grapheme ‘ӱ’. Any difference in language varieties should also be accounted for, although these are believed to be relatively minor (Krauss, 1975).

7. Conclusion

The postbase attachment process presented herein for Yupik may have implications for morphophonological theory, in that it lends credence to the idea that derivational morphology must be performed cyclically in some languages in order to derive the proper surface form (§ 2.2.). In particular, it suggests that the underlying string of an utterance is processed in phases rather than across the full string at once. In discussing the implementation of the Yupik morphological analyzer however, we have presented a design technique to handle this phenomenon, and anticipate that usage of the Foma toolkit in this way may be adapted to languages that require such a processing pattern, for instance, other languages in the Inuit-Yupik language family, which are the most typologically similar to Yupik. Nevertheless, while much work remains to be explored in this respect and in regards to efficacy evaluation, our implementation incorporates all the lexical items and morphophonological rules of Yupik that have been documented and described to date.

8. Bibliographical References

Badten, L. W., Kaneshiro, V. O., Oovi, M., and Koonooka, C. (2008). *St. Lawrence Island / Siberian Yupik Eskimo Dictionary*. Alaska Native Language Center, University of Alaska Fairbanks.

Beesley, K. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications, Stanford, CA.

Bills, A., Levin, L. S., Kaplan, L. D., and MacLean, E. A. (2010). Finite state morphology for Iñupiaq. *7th SaLTMil Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages*, pages 19–26.

de Reuse, W. J. (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.

Hulden, M. (2009a). Fast approximate string matching with finite automata. *Procesamiento del Lenguaje Natural*, 43:57–64.

Hulden, M. (2009b). 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.

Hulden, M. (2011). Morphological analysis tutorial: A self-contained tutorial for building morphological analyzers. `https://fomafst.github.io/morphtut.html`. Accessed: 2017-09-19.

Institute for Information Technology. (2012). The UQAILAUT Project: Inuktitut Morphological Analyzer. http://inuktitutcomputing.ca/Uqailaut/info.php.

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

Koonooka, (Petuwaq), C. (2005). Yupik language instruction in Gambell (St. Lawrence Island, Alaska). *Études/Inuit/Studies*, 29(1/2):251–266.

Krauss, M., Holton, G., Kerr, J., and West, C. T. (2010). Indigenous peoples and languages of Alaska. ANLC Identifier G961K2010.

Krauss, M. (1975). St. Lawrence Island Eskimo phonology and orthography. *Linguistics: An International Review*, 13(152):39–72, January.

Krupnik, I. and Chlenov, M. (2013). *Yupik Transitions — Change and Survival at Bering Strait, 1900-1960*. University of Alaska Press, Fairbanks, Alaska.

Schwalbe, D. M. (2017). Sustaining linguistic continuity in the Beringia: Examining language shift and comparing ideas of sustainability in two Arctic communities. *Anthropologica*, 59(1):28–43.