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. 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, /a/, /i/, /u/ which can be lengthened (with the exception of /a/) 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 ghbw, m and mm, n and nn, ng and nngng, ngw and nngw.

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 +ug (to want one to V) which yields kavitesug- (to want one to be red), most postbases trigger a series of morphophonological changes at the
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 morphophonological 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 @~ -rągkiigh (to V quickly). Modification of base-final te, which is represented by the symbol @, occurs before base-final e is dropped via ~. Otherwise, the ordering of the symbols is arbitrary, since the morphophonological processes being represented are mutually exclusive, as in ~ 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 @~ -rąghkiigh is between @ and ~. 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. (ng) If the base ends in a vowel, affix ng
2. ~ If e appears in base-final or penultimate (semifinal) 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 aghnu- (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 ~-(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. ~ 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- ~-(g/t)u-q, we derive aghnaaguq.

2.3. Derivation of a More Intricate Word

Here, we derive a Yupik word of greater complexity, pa-
gunghalighnaqaqa (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 pagunghag*

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

1. aghnaaguq
   aghnagh- ~-(ng)u- ~-(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 (semifinal) 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 aghnu- (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 ~-(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. ~ 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- ~-(g/t)u-q, we derive aghnaaguq.

2.3. Derivation of a More Intricate Word

Here, we derive a Yupik word of greater complexity, pa-
gunghalighnaqaqa (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 pagunghag*

1. (g/t) If the base ends in a vowel or a consonant, affix allomorphs g or t respectively
2. ~ 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- ~-(g/t)u-q, we derive aghnaaguq.
(2) Naagpek sagnegha
    naagpek  sagnegha
  naa~gpek-~saghnegh~(ng)a-
mother-2SG.POSS bowl-3SG.POSS

pagunghalighnaqaqa
pagunghagh*---ligh-~@~naqe-~(g)a--qa
crowberry-→to.put.N.in-→to.be.going.to.V-→TRANS.IND→1SG.3PL
(I am going to put crowberries in...)

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 lexic 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 (§4).

4. Implementation

The Yupik analyzer is strictly implemented within the lexc and foma languages, although lexical transducers for sister languages to Yupik such as Inupiaq have incorporated other resources such as XML databases (Bills et al., 2010), and an analyzer for Inuktut 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 aghaagnaq are displayed in Figure 2. The noun base aghaagn 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)a[U→V]. The string derived thus far then continues to the VInf1 continuation class, and is concatenated with the underlying gloss of the inflectional
endings, [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 ~\%(g/t)uq respectively, for processing in foma.

The noun-elaborating postbase ~ghghagh 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- ~ghghagh- ~\%(ng)u- ~\%(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 ~\%(g/t)u, while the analogous ending for transitives is ~\%(g/\t)u.

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.TRNS@

Strings that contain morphemes with mismatched diacritics are discarded, as demonstrated in the expanded lex file presented in Fig. B. For instance, transitive verb ungapate (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 ungapate 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.

### 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 C. Rules take the form \( A \rightarrow B || \Gamma, \Delta \), where \( A \) is rewritten as \( B \) in the context of \( \Gamma \) and \( \Delta \). In the Yupik analyzer, \( \Delta \) typically refers to the substring that is rewritten as \( B \) in the morphophonological context determined by \( \Gamma \) and \( \Delta \), where \( \Delta \) 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 combines the rewrite rules together in the
read lexc example.lexc

::
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 .o.
  "(g/t)" -> 0;
::
define UvularDropping
  "gh" -> 0 || [V - e] _ " : " [V - e] .o.
  ":" -> 0;
::
define Grammar
  Lexicon .o.
  InsertMBndry .o.
  ResolveAllomorphy .o.
  SemiAndFinalE .o.
  UvularDropping .o.
  FinalE .o.
  VowelDominance .o.
  CleanupMBndry .o.

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 *e* 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 *aghnaqh*j:*(...)=*:(ng)*j:*...(g/t)*j:*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 intro-

duced 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.

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

read lexc example.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 _;
::

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

While the reference grammar was instrumental to the implementation of the analyzer, the grammar nonetheless 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 lex 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 of 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 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.

![Figure 6: Of the Yupik word types in the 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.](image)

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.

<table>
<thead>
<tr>
<th></th>
<th>Types</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>97.82</td>
<td>97.61</td>
<td>97.46</td>
</tr>
<tr>
<td>Tokens</td>
<td>98.20</td>
<td>97.11</td>
<td>97.90</td>
</tr>
</tbody>
</table>

The iterative process first described in § 2.2. From the underlying string aghnah^\sim/.(ng)u^\sim/(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.
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 5 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 inflection as a series of eleven “paradigmatic sets”, and within each set, demonstratives take on an additional six flection as a series of eleven “paradigmatic sets”, and within each set, demonstratives take on an additional six

### 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 (§ 4.2). 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


