Finite-State Morphology for Iñupiaq

Aric Bills[1], Lori S. Levin[2], Lawrence D. Kaplan[3], Edna Ahgeak MacLean[4]

[1,3,4]Alaska Native Language Center, [2]Language Technologies Institute
[1,3,4]University of Alaska Fairbanks, [2]Carnegie Mellon University
[1]aric.bills@gmail.com, [2]lsl@cs.cmu.edu, [3]ldkaplan@alaska.edu, [4]edna.maclean@gmail.com

Abstract

This paper describes ongoing work to develop a finite-state computational morphology of North Slope Iñupiaq, an indigenous North American language with exceptionally productive derivational morphology, complex inflection, and considerable morphologically conditioned phonological phenomena. The language-independent Xerox Finite-State Tools serve as the underlying engine for our lexical transducer and ultimately make this project possible, but a language-specific abstraction layer implemented above the *lexc* finite-state lexicon definition language has made it possible to develop the morphology more quickly and in a more natural way, which we believe will lead to improved maintainability and scalability.

1. Introduction

This paper describes ongoing work to develop a computational morphology of North Slope Iñupiaq, a language with exceptionally productive derivational morphology, complex inflection, and considerable morphologically conditioned phonological phenomena. Our work expands on earlier work on Iñupiaq computational morphology by Per Langgård and Trond Trosterud. In this paper we wish specifically to draw attention to ways in which a format tailored to the nature of Iñupiaq morphophonology has made this process simpler and more natural, and therefore, we hope, easier to maintain and expand.

2. Iñupiaq

Iñupiaq is an Eskimo-Aleut language and the westernmost member of the Inuit dialect continuum, which extends from northern Alaska to Greenland. With approximately 2144 living speakers from a community of 15,700 (Krauss, 2007, p. 409), Iñupiaq is considered endangered. The present work concerns the North Slope dialect, which is spoken in the Alaskan villages of Kivalina, Point Hope, Point Lay, Wainwright, Atqasuk, Barrow, Nuiqsut, Kaktovik, and Anaktuvuk Pass (MacLean, 1986a, p. x). North Slope Iñupiaq exhibits both lexical and phonological differences from other dialects. Among other differences, it is the only dialect which has all of the following: a series of palatal consonants; surface clusters of vowels of different qualities (that is, vowel clusters in addition to phonetically long vowels); and no consonant clusters that do not agree in terms of voicing (i.e., both consonants are voiced or neither are voiced) and continuancy (i.e., both are obstruents or neither are obstruents).

North Slope Iñupiaq, like all Eskimo languages, is highly polysynthetic and has a elaborate inflectional system. Its phonology is generally more conservative and more complex than Canadian and Greenlandic Inuit dialects. Most suffixes trigger morphophonological alternation at morpheme boundaries; additionally, "a great many Inupiaq suffixes exhibit allomorphy for which no one proposes a synchronic phonological account" (Kaplan, 1981, p. 232).

Iñupiaq examples in this paper will be given in the standard orthography.[1]

2.1. Iñupiaq morphology

Iñupiaq grammatical categories include nouns, verbs, demonstratives, personal pronouns, and particles. Of most interest morphologically are nouns and verbs, which allow both extensive derivational morphology and complex inflection; demonstratives also allow rich inflection as well as limited derivational morphology.

The basic structure of an Iñupiaq noun, verb, or demonstrative is *base + zero or more postbases (bound derivational suffixes) + inflectional ending + zero or more enclitics*. The process of postbase attachment may be considered a recursive, stem-deriving process; a stem may be defined as either a base or a stem plus a postbase. The main morphotactic constraint on Iñupiaq stems (aside from semantic considerations, which will not be taken into account here) is that postbases and inflectional endings must match the category of the stem to which they attach; in other words, nominal suffixes attach to nominal stems, verbal suffixes attach to verbal stems, etc. Postbases which attach to a particular category may derive stems of a different category; for example, postbase *-qaq-* 'have' attaches to nominal stems and derives verbal ones, as in *qamutiqaqtuŋa* 'I have a car', from *qamun* 'car' + *-qaq-* 'have' + *-tuŋa* (indicative present 1st person singular).

Nouns are inflected for case and grammatical number (singular, dual, or plural) and for grammatical person and number of their possessor, if any. Verbs are inflected for mood and grammatical person and number of their subject, as well as grammatical person and number of any definite direct object. Demonstratives may be inflected as pronouns, in which case they are inflected for case and the grammatical number of their antecedent, or as adverbs, in which case they are inflected for case only. Verbal inflections are explicitly transitive or intransitive, so an additional, long-distance morphotactic constraint is that verbal inflection be compatible with stem valence. Additionally, some noun stems are restricted

[1]For a guide to pronunciation, see `http://www.alaskool.org/Language/inupiaqhb/Inupiaq_Handbook.htm`.

Kepa Sarasola, Francis M. Tyers, Mikel L. Forcada (eds.)
7th SaLTMiL Workshop on Creation and Use of Basic Lexical Resources for Less-Resourced Languages, LREC 2010, Valetta, Malta, 23 May 2010, p. 19–26

as to the grammatical numbers for which they may be inflected; for example, *kamikłuuk* 'pants' cannot bear singular inflection.

Enclitics are words which are syntactically distinct from other words but phonologically (and orthographically) bound to the previous word. A distinction can be made between "reduced forms"—enclitics which have full-word counterparts—and "true" enclitics, which have no such counterparts. An example of a reduced form is *=una* 'this' as in *sunauna* 'what is this?'; *una* may also occur as an independent word, as in *una qimmiq siñiktuq* 'this dog is sleeping'. In contrast, a "true" enclitic such as *=lu* 'and' as in *tuttulu qimmiġlu* 'the caribou and the dog' cannot be separated from the words to which it attaches; **tuttu lu qimmiq lu* is ungrammatical.

2.2. Iñupiaq morphophonology

Different suffixes in Iñupiaq trigger different morphophonological alternations at their left boundaries, and the pattern of alternations a suffix will trigger is not entirely predictable from the phonetic form of the suffix. For example, postbase *-saġataq-* 'for a long time' attaches directly to stems without deleting anything; postbase *-sugruk-* 'a lot' deletes stem-final consonants; and postbase *-siññaq-* 'only' deletes stem-final /t/ but not /k/ or /q/. Other patterns include deleting penultimate /i/, deleting stem-final back consonants, or deleting stem-final syllables. Some suffixes trigger gemination of the onset of the preceding syllable. Edna MacLean, in her Iñupiaq pedagogical materials (1986a; 1986b; unpublished), indicates the morphophonological attachment pattern of a suffix with one of eight symbols; for example, '-' indicates that a stem-final consonant is deleted, while '+' indicates that no stem-final segment is deleted.

Many suffixes also have phonologically conditioned allomorphs, and different morphemes are sensitive to different environments. For example, postbase *-tiq-* 'quickly or abrubtly' becomes *-liq-* when preceded by a vowel (suffix *-tuksrau-* 'must' becomes *-ruksrau-* in the same environment); postbase *-suk-* 'want to' becomes *-uk-* when preceded by a back consonant; absolutive plural marker *-t* becomes *-it* after a /k/ or certain lexically conditioned instances of /q/, and before other instances of /q/ it optionally triggers gemination.

3. Finite-State Morphology

Finite-state transducers are directed graphs representing rational relations between sets of strings, and are an elegant way to model morphology computationally (Beesley, 2004b, p. 3). They are compact, fast, and inherently bidirectional (meaning that a single morphological transducer can be used equally well for generation as for analysis). A lexical transducer is a transducer that maps surface forms of words onto abstract, morphologically decomposed "underlying" forms[2], by combining a lexicon (consisting of underlying forms of words) together with a set of phonological (or, more accurately, graphemic) rules (Karttunen et al., 1992).

The Xerox Finite State Toolkit (XFST) is probably the most widely used software for creating lexical transducers (Kornai, 1999, p. 4). It provides two languages, *xfst* and *lexc*, designed to be used in tandem. *xfst* is a language with a rich calculus for specifying regular expressions, most commonly used to model phonological rewrite rules. *lexc* is a right-recursive phrase-structure grammar (Beesley and Karttunen, 2003, p. 203) for specifying lexicons in an underlying form via morpheme concatenation. Because a grammar based on concatenation alone cannot easily restrict the co-occurrence of non-adjacent morphemes within a word, *lexc* also provides a mechanism called a "flag diacritic." Flag diacritics set or query memory registers and can be associated with specific morphemes. Any word containing two morphemes with incompatible flag diacritics is effectively filtered out of the lexicon (Beesley and Karttunen, 2003, pp. 339–373).

4. Langgård and Trosterud's transducer

Per Langgård of Okaasileriffik (the Greenland Language Secretariat) and Trond Trosterud of the University of Tromsø have developed a proof-of-concept Iñupiaq transducer,[3] and generously furnished us with the XFST source code at the beginning of our project. We referred to this code frequently in the early stages of development of our transducer and incorporated several key features from it, including the use of the symbol '>' as a morpheme boundary marker and the definition of sets of characters (vowels, plosives, voiced fricatives) to be used for convenience in morphographemic rules. Two additional techniques adopted from this transducer are especially pertinent to the discussion at hand: first, flag diacritics are used to ensure that verb inflections reflect the valence of their stem (in other words, that intransitive-only verb stems not be inflected with transitive endings); and second, extensive use is made of "rule triggers"—special tags attached to morphemes to indicate that the morpheme conditions a particular alternation (see Uí Dhonnchadha, 2003, p. 46).

5. Implementation

5.1. Morphographemics

Productive phonological processes are modeled using morphographemic rewrite rules written in *xfst*. Langgård and Trosterud's rules were rewritten to correspond more closely to Edna MacLean's analysis of Iñupiaq phonology, to facilitate the inclusion of lexical material from her work.[4] In particular, each of MacLean's suffix combination patterns (see Section 2.2.) was implemented as a cascade of rules sensitive to the presence of a specific rule trigger, which is deleted as the last step in the cascade. Rules also exist for the

[2]For example, a lexical transducer for English might map the surface form *hidden* onto the underlying form *hide+PastParticiple*; one for Iñupiaq might map surface form *tautuŋniaġiga* 'she/he will see me' onto underlying form *tautuk+niaq+IndicativePresent+3Sg+1SgObj*.

[3]`http://giellatekno.uit.no/ipk.html`

[4]Langgård and Trosterud's transducer is based on the Greenlandic tradition of Eskimo analysis, with which the first author was unfamiliar and which would have made it more difficult to use MacLean's work; rewriting the rules also allowed the first author to come to grips with *xfst*. The rewriting was not due to any inaccuracy in Langgård and Trosterud's code.

formation of the absolutive dual stem (which serves as the basis for several other dual forms), demonstrative-specific alternations, gemination, palatalization, assimilation, and the conversion between the transducer-internal format and standard Iñupiaq orthography.

5.2. Lexicon

The lexicon is defined in a series of text files whose format was designed to optimize data entry; the contents of these files are converted first to XML, then to *lexc* format. The underlying data model of the source files is compatible with the phrase-structure grammar of *lexc*, but the format of the files themselves is quite different. In *lexc*, files are structured as lists of word formatives called LEXICONs. Each member of a LEXICON can specify a "continuation class"—another LEXICON whose members it will accept as suffixes. Members of a LEXICON may be empty strings, in which case the members of the specified continuation class essentially become members of the empty string's LEXICON.

Beesley (2004b; 2004a; 2003) advises against creating *lexc* lexicons from scratch, calling them a "dead-end" (Beesley, 2004a, p. 2) because the format is specific to the Xerox Finite-State Tools and the data are too sparse to be very useful to other applications. Instead, Beesley recommends creating lexical resources in XML, which can be used to represent data sets of arbitrary complexity. However, writing XML by hand is cumbersome and error-prone. As a compromise, we have created a set of lightweight, text-based formats designed to allow us to enter essential information about each morpheme in a quick, natural way, and a Tcl script to convert this information into XML; from that format it is then converted to *lexc* format.[5] At present, the XML representation of the lexical data contains very little information other than what is needed to build a lexical transducer, but others who require Iñupiaq lexical data should find it reasonably easy to use and expand upon. Because the XML itself is rather unremarkable, no more will be said about it in this paper.

Separate source files exist for bases, postbases, inflectional suffixes, and enclitics. Each file begins with a metadata section where shorthand morpheme category codes are defined and associated with LEXICON names, continuation classes, and flag diacritics.[6] With the exception of the inflectional suffix file, morphemes are then listed in the order in which they appear in the dictionary or grammar book, without any special grouping by category. In the stem file, each entry consists of an orthographic form followed by a category code; one may optionally specify one or more of the following: separate lexical (underlying) and surface forms (separated by a colon), a comment (beginning with an exclamation point), an English-language gloss (beginnning with a hash mark), and a reduced form of the stem (beginning with a tilde). Figure 1 presents sample stem entries from a variety of categories: *aaġlu* 'killer whale' is a noun stem; *aasii* 'and [then]' is a conjunction; *ikayuq-* 'help' is a verb stem which may be either intransitive (e.g., *ikayuqtuq* 'he/she is helping') or transitive (e.g. *ikayuġaaŋa* 'he/she is helping me'); *iraqtu-* 'be wide' is an intransitive-only verb stem (e.g. *iraqturuq* 'it is wide'); *ñiaq* 'don't do that!' is an interjection; and *suna* 'what' is an interrogative pronoun. The entry for *suna* shows how one specifies separate lexical and surface forms; surface form *suna* is specified here as a special absolutive singular form of the stem *su-* ('what').

```
aaġlu n # killer whale
aasii conj ~asii # and [then]
ikayuq it # to help [someone]
iraqtu i # to be wide
ñiaq interj # don't do that
su>+Pro+Abs+Sg:suna pro # what
     ! interrogative pronoun
```

Figure 1: Example stem entries.

In the postbase and enclitic files, each entry includes an orthographic form, a membership category code (denoting the class to which the morpheme belongs; these include 'n' [noun], 'i' [intransitive verb], 't' [transitive verb], and 'it' [ambitransitive verb]), and a continuation category code (specifying the morpheme's continuation class; in addition to the membership category codes, code 'same' marks verb-attaching postbases which derive verbs of the same valence, whatever that may be). Entries may optionally specify English-language glosses, comments, and separate lexical and surface forms. Example postbase definitions are given in Figure 2; sample words containing these postbases are given in examples 1–6, which will be discussed below. All examples are from the personal files of Edna MacLean unless otherwise noted.

```
+ġruiññaq n n # merely, only, just a
-qaq n i # have
+-qasIq i t # to V at the same time with Obj
{?C -+sima ?V +ma} it same #it is now known that
+[s]uk it same # want to
+t//liq it same # quickly, abruptly
```

Figure 2: Example postbase entries.

(1) *iqalugruiññaq*
iqaluk-ġruiññaq
fish-merely
'merely a fish'

[5] An anonymous reviewer points out that there are XML editors which allow data to be entered as simply as with the non-XML formats described here.While our format allowed lexical data to be entered easily and represented in a natural way, this could and probably should have been done directly in XML.

[6] We adopt Langgård and Trosterud's practice of using flag diacritics to enforce valence restrictions on verb stems; additionally, we use them to enforce grammatical number restrictions on certain noun stems, such as *kamikłuuk* (see Section 2.1.).

(2) *kamiqaqtuŋa*
kamik-qaq-tuŋa
boot-have-IND.PRS.1SG
'I have boots/a boot' or 'I am wearing boots' (MacLean, 1986a, 50)

(3) a. *savaqasiġaa*
savak-qasiq-kaa
work-at.same.time.with-IND.PRS.3SG>3SGOBJ
'he/she is working with her/him'
b. *aqpatqasiqsaġa*
aqpat-qasiq-taġa
run-at.same.time.with-IND.PST.1SG>3SGOBJ
'I ran with her/him'

(4) a. *naviksimaruaq*
navik-sima-tuaq
break-it.is.now.known-IND.PST.3SG
'it did break'
b. *naatchimaruat*
naatchi-sima-tuat
finish-it.is.now.known-IND.PST.3PL
'they did finish'

(5) a. *ilausukpit*
ilau-suk-pit
be.included-want.to-INT.2SG
'do you want to be included?'
b. *tautugukkiga*
tautuk-suk-kiga
see-want.to-IND.PRS.1SG>3SGOBJ
'I would like to see it'

(6) a. *naviktiqtuq*
navik-tiq-tuq
break-quickly-IND.PRS.3SG
'it broke instantaneously'
b. *ikuliqtuq*
iku-tiq-tuq
get.into-quickly-IND.PRS.3SG
'he/she quickly got in [e.g., a car]/on [e.g., an airplane]'

In the lexical data files, surface forms of suffixes (postbases, inflections, and enclitics) begin with a rule trigger symbol indicating the morphophonological alternation pattern conditioned by the suffix (see Sections 2.2. and 5.1.). For example, symbol '+' indicates that no stem-final segments are deleted; if the suffix begins with two consonants and is affixed to a stem ending in a consonant, the first consonant of the suffix is deleted. Thus, in example 1, suffix-initial *ġ* is deleted and stem-final *k* remains, becoming *g* due to assimilation with the following *r*. Symbol '-' indicates that any stem-final consonant is deleted; this can be seen in example 2, where the *k* of *kamik* is deleted. Symbol '+-' indicates that stem-final *k* or *q* is deleted (see example 3.a), but not stem-final *t* (see example 3.b).
Many postbases and inflectional endings have multiple phonologically conditioned allomorphs. These are specified within curly braces as a list of alternating "condition codes" and forms or form lists (a form list is enclosed within an additional pair of curly braces). In Figure 2, postbase *-[si]ma-* 'it is now known that' is defined; condition code `?C` indicates that allomorph *-sima-* occurs following a consonant (as in example 4.a); condition code `?V` indicates that allomorph *-ma-* occurs after a vowel (as in example 4.b). When allomorphs are specified, the first allomorph listed will be used as the lexical form of all allomorphs, so that all allomorphs are analyzed as the same morpheme. Shorthand notation exists for two common allomorphy patterns, eliminating the need for curly braces or condition codes. The notation `[C]` (e.g., `+[s]uk` 'want to') indicates that the bracketed consonant appears following a vowel or /t/ (see example 5.a) and is omitted otherwise (see example 5.b; note that the stem-final *k* of *tautuk-* becomes *g* due to assimilation). The notation `C//C` (e.g., `+t//liq` 'quickly, abruptly') means that the allomorph beginning with the consonant to the left of the double slash (in this case, *-tiq-*) is used if the preceding segment is a consonant (see example 6.a); otherwise the allomorph with the consonant to the right of the double slash (here, *-liq-*) is used (see example 6.b).
Inflectional endings are specified in two-dimensional "tables." An example table implementing unpossessed and possessed absolutive singular noun endings is given in Figure 3.

```
Table n +N {
  Columns {
    {}
    +1Sg  +1Du  +1Pl
    +2Sg  +2Du  +2Pl
    +3Sg  +3Du  +3Pl
    +3RSg +3RDu +3RPl
  }
  Row +Abs+Sg {
    {}
    {?V +ga ?C +a} +kpuk +kput
    {?kQ :iñ ?Otherwise -n} +ksik +ksi
    {?Always -ŋa ?notVthenV :a}
        {?Always -ŋak ?notVthenV :ak}
        {?Always -ŋat ?notVthenV :at}
    -nI +ktik {?Always {+ktiŋ -riŋ}}
  }
}
```

Figure 3: Absolutive singular inflectional suffixes defined as a table.

The keyword `Table` signals the beginning of a new table definition; this is followed by a category code to be associated with each suffix in the table, and a string of grammatical tags which apply to the table as a whole. In Figure 3, the category code is '`n`' and the grammatical tag string is `+N`. The rest of the table definition is enclosed in curly braces. A table contains exactly one `Columns` declaration and one or more `Row` declarations. The `Columns` declaration specifies one string of grammatical tags for each column in the table. In the example table, the columns in the table correspond to grammatical possessors; the first column is for unpossessed forms, and thus the grammatical tag string for this

Case & number	*Possessor* none	1Sg	1Du	1Pl	2Sg	2Du	2Pl	3Sg	3Du	3Pl	3RSg	3RDu	3RPl
Abs. Sg.	∅	+ga[a], +a[b]	+kpuk	+kput	:iñ[c], -n[d]	+ksik	+ksi	-ŋa[e], :a[f]	-ŋak[e], :ak[f]	-ŋat[e], :at[f]	-nI	+ktik	+ktiŋ[e], -riŋ[e]

[a]used with vowel-final stems; [b]used with consonant-final stems; [c]used with stems ending in a strong consonant (*k* and some *q*); [d]used with stems ending in a vowel or a weak consonant (*t* and some *q*); [e]can be used in any phonological context; [f]more conservative form; cannot be used with stems ending in a consonant cluster

Figure 4: Contents of Figure 3 presented as a row in an inflection table.

	Possessor number		
Possessor person	Singular	Dual	Plural
no possessor	∅		
1st	+ga[a], +a[b]	+kpuk	+kput
2nd	:iñ[c], -n[d]	+ksik	+ksi
3rd	-ŋa[e], :a[f]	-ŋak[e], :ak[f]	-ŋat[e], :at[f]
3rd reflexive	-nI	+ktik	+ktiŋ[e], -riŋ[e]

(see Figure 4 for footnotes)

Figure 5: Alternative representation of Figure 3 row contents as a two-dimensional table.

```
LEXICON NounInflection
+N+Abs+Sg:0 Enclitics
+N+Abs+Sg+1Sg:%?V%+ga Enclitics
+N+Abs+Sg+1Sg:%?C%+a Enclitics
+N+Abs+Sg+1Du:%+kpuk Enclitics
+N+Abs+Sg+1Pl:%+kput Enclitics
+N+Abs+Sg+2Sg:%?kQ%:iñ Enclitics
+N+Abs+Sg+2Sg:%?Vt%-n Enclitics
+N+Abs+Sg+2Du:%+ksik Enclitics
+N+Abs+Sg+2Pl:%+ksi Enclitics
+N+Abs+Sg+3Sg:%-ŋa Enclitics
+N+Abs+Sg+3Sg:%?notVthenV%:a Enclitics
+N+Abs+Sg+3Du:%-ŋak Enclitics
+N+Abs+Sg+3Du:%?notVthenV%:ak Enclitics
+N+Abs+Sg+3Pl:%-ŋat Enclitics
+N+Abs+Sg+3Pl:%?notVthenV%:at Enclitics
+N+Abs+Sg+3RSg:%-nI Enclitics
+N+Abs+Sg+3RDu:%+ktik Enclitics
+N+Abs+Sg+3RPl:%+ktiŋ Enclitics
+N+Abs+Sg+3RPl:%-riŋ Enclitics
```

Figure 6: One possible representation of the contents of Figure 3 in *lexc* format.

column is empty. Each `Row` declaration specifies a string of grammatical tags that apply to that row followed by a list of the surface forms of the suffixes in that row. The row in Figure 3 defines absolutive singular endings and is accordingly tagged `+Abs+Sg`. In the actual inflection file, the table of noun inflections contains 24 rows, one for each possible combination of case and grammatical number, but due to space constraints only one row is reproduced in Figure 3. Conceptually, the contents of this figure correspond to the table shown in Figure 4. Thinking in terms of a thirteen-column table can be daunting; we have dealt with this challenge by strategically inserting white space and newlines in both the column list and the row contents, as can be seen in Figure 3. This extra space is ignored by the software converting the tables to XML and *lexc*, but allows humans editing the file to visualize each row in terms of a more compact table, such as the one presented in Figure 5.

Like postbases, inflectional suffixes may exhibit allomorphy, and the same notation used for postbases with allomorphs is used in inflection tables. The special condition code `?Always` is used to denote variants which are not phonologically conditioned, and the code `?Otherwise` indicates that an allomorph occurs in all environments where no other allomorphs occur. Condition code `?kQ` specifies an allomorph that attaches to stems ending in *k* or "strong" *q*.[7] Condition code `?notVthenV` prohibits an allomorph from attaching to a stem ending in a vowel cluster.

The lexical form of each inflectional ending is the concatenation of table tags + row tags + column tags. For example, in Figure 3, the tag string `+N+Abs+Sg+1Pl` (absolutive singular noun with first person plural possessor) corresponds to the suffix `+kput`.

The table mechanism provides a practical alternative to representing inflectional information directly in *lexc*, as for example in Figure 6. The *lexc* representation involves considerable redundancy, both in the tag strings and in the continuation classes (although other langauges might require a more complex continuation class structure than the one shown here). On a more subjective note, we believe the table structure is easier to read and (assuming one is entering inflectional data from tabular printed sources) considerably easier to write.

5.3. Condition codes

The specification of allomorphic variants in the source files is simple enough, but the back-end implementation of this feature is somewhat more complex. Allomorphic variants might be handled within *lexc* by creating separate LEXICONs for morphemes belonging to each conditioning environment and implementing continuation classes that respect the restrictions associated with each environment. For example, vowel-final verb stems and postbases would continue

[7]Some instances of *q* at the end of noun stems are considered "strong" and interact with certain suffixes in the same way as *k* and differently from how "weak" instances of *q* interact with the same suffixes. The distinction between strong and weak *q* is partially conditioned by phonological factors and partially an idiosyncratic attribute of specific stems. In all cases, the morphemes sensitive to this distinction are noun inflection suffixes.

to a LEXICON containing allomorphs sensitive to that environment and excluding allomorphs which attach only to consonant-final stems. This approach is further complicated by the interaction of different conditioning environments. For example, the imperative singular intransitive ending has allomorphs sensitive to the following environments: stems of the form (C)VCV-, stems ending in *k* or *q* or consisting of more than two syllables and ending in a vowel, stems ending in *t*, and stems ending in a two-vowel cluster. Stems of the form (C)VCV would need to continue to a class containing not only suffixes sensitive to the form (C)VCV-, but also suffixes conditioned by a preceding vowel, and suffixes without conditions. Stems ending in a two-vowel cluster would need to continue to a distinct LEXICON containing suffixes conditioned by -VV-, suffixes conditioned by a preceding vowel, and suffixes without conditions. It should be clear that handling phonologically conditioned allomorphy via the architecture of a *lexc* grammar would require a complex maze of LEXICONs and continuation classes.

Fortunately, *xfst* offers an elegant alternative which leverages its pattern-matching strengths. In the script which converts source files into *lexc* format, each allomorph is tagged with a rule trigger corresponding to its condition code. LEXICONs are then constructed on the basis of morphotactics alone, without phonological considerations. A lexicon defined in this way will overgenerate, attaching suffix allomorphs to stems regardless of whether those stems fulfill the requisite phonological conditions. To address this problem, the lexicon is filtered through a series of rules, defined in a separate file, which are sensitive to the condition code triggers; these rules eliminate any string where the characters preceding the trigger do not match a specified pattern. After accepting a string, the rules remove the trigger from the string. This approach allows a clean separation between morphotactic and phonological constraints.

Since this filtering mechanism requires us to write a number of *xfst* rules, one might wonder whether it would not be simpler to create a set of rewrite rules to produce the appropriate allomorphs directly. For some languages (particularly with language documentation written in a certain style), this may be the best approach. In our case, this approach would have two considerable drawbacks. First, it is unlikely that the set of rules required to produce all allomorphs could be smaller than the set of filter rules currently in place; in addition to specifying specific alternations, rewrite rules would need to implement the environments currently specified in the filter rules, and in many cases a single filter would need to correspond to multiple rewrite rules. Second, the filter system is a natural implementation of the way allomorphs are specified in MacLean's Iñupiaq language materials (1981; 1986a; 1986b; unpublished), which serve as the bulk of our source material; translating this into a set of rewrite rules would have required significant additional work.

Two particularly important comments from an anonymous reviewer deserve to be addressed here. First, although we have treated Iñupiaq inflection in terms of phonologically conditioned allomorphs, it is also possible to conceive of it in terms of lexically conditioned inflection classes. The strongest evidence for this analysis is the fact that, for some inflectional endings, a different form is used for stems ending in "strong *q*" than for stems ending in "weak *q*" (see footnote 7). The treatment of inflection classes is commonly and properly done in *lexc*, and this is the approach taken by Langgård and Trosterud. On the other hand, although strong and weak *q* might not properly be considered distinct phonemes, it is trivial and unproblematic to treat them as if they were, and having done so, allomorphs of inflectional endings may be chosen entirely on the basis of the preceding (pseudo)phonological environment. The majority of Iñupiaq inflectional endings have a single allomorph (or a set of allomorphs produced entirely from fully productive phonological processes), so for Iñupiaq, the inflection class treatment would necessitate a large amount of duplication, which can be avoided by treating different forms of inflectional endings as phonologically-conditioned variants. The reviewer's other criticism is that any advantage that the filter approach may have over a LEXICON/continuation class approach in terms of elegance must be accompanied by a concomitant decrease in performance, and that this is performance hit will become more severe as the lexicon increases in size. This is certainly true at compile time; although we have no hard numbers, when we switched from a *lexc*-based approach to the filter approach, we noticed that compilation took perhaps two minutes when before it was well under 30 seconds (on a modest computer built in 2005). However, we do not notice or expect there to be an important difference in performance at runtime, since in both cases the end result of compilation is a highly optimized finite-state transducer the traversal of which is straightforward. The primary problem then, compile time, is an issue for developers but not for end users. Although Moore's Law[8] cannot make this problem go away, it does suggest that the impact on developers will diminish over time. When considering performance issues, one must also bear in mind that the *lexc* approach may be considered to impose a performance hit during development, in that the construction of appropriate LEXICONs and continuation classes requires the developer to perform by hand (or accomplish in some other automated way, which would also take time) the filtration which in our system is done by *xfst*.

6. Current status and future plans

Currently, the North Slope Iñupiaq transducer implements most of the lexical, morphotactic, morphophonological, and inflectional information contained in Edna MacLean's *Abridged Iñupiaq and English Dictionary* (1981) and three-year university-level Iñupiaq curriculum (1986a; 1986b; unpublished), as well as most North Slope Iñupiaq entries from Donald H. Webster and Wilfried Zibell's *Iñupiat Eskimo Dictionary* (1970). We are in the process of adding additional stems and postbases from the private files of Edna MacLean.

At present, the transducer does not attempt to handle proper nouns or recent loanwords, which are subject to slightly different morphophonological rules (MacLean, 1986a, pp. 154–155). More work also remains to be done to expand the

[8] Moore's Law predicts that the number of transistors on an integrated circuit will double roughly every two years. The practical implication of this trend is that new computers are consistently and increasingly more performant than their predecessors.

set of stems and postbases in the lexicon. We are working to develop a stem-guessing transducer (Beesley and Karttunen, 2003, pp. 445–448) which may help with that process; a postbase guesser is also not out of the question.

The transducer has been informally tested against a corpus of texts for university-level second language learners of Iñupiaq. At present, the transducer generates at least one analysis for 3414 out of 4406 tokens (77.49%) and 2021 out of 2887 unique types (70.00%). So far, no systematic attempt has been made to evaluate the accuracy of the analyses produced by the transducer. An additional 5000 words, unseen by the developers, have been set aside for additional testing.

As the transducer becomes more mature and able to recognize more words, we hope to incorporate it in additional technologies that may benefit the Iñupiaq community. In particular, we hope to develop a spell-checker, which Iñupiaq language learners have expressed an interest in, and Iñupiaq-aware OCR software (Yoo, 2008) to help the community digitize existing materials in text format. We would also like to develop tools geared toward the academic community, such as a lemmatizer. The architecture of the transducer could also be easily and naturally applied to other dialects of Iñupiaq and probably to languages within the Yupik branch of the Eskimo language family as well.

7. Conclusions

This paper describes some of the specific challenges presented by the Iñupiaq language which a computational morphology must address, and how we have dealt with those points by creating language-specific formats for representing lexical and morphophonological information. Our system provides customized treatment for each of the fundamental morpheme types found in Iñupiaq: bases, postbases, inflectional endings, and enclitics. In these formats, data need not be regrouped according to grammatical category or phonological structure; it can be entered in the order in which it appears in the dictionary. Inflectional endings are specified in a two-dimensional format corresponding more closely to the way linguists conceive of inflectional paradigms. Special mechanisms have been developed to handle allomorphy and long-distance dependencies (valence restrictions on verbs and number restrictions on certain nouns) in a natural way.

While the formats used for lexical data in this project are probably geared too specifically to Iñupiaq to be reused for any but the most closely related languages, the concept of defining data in a convenient, language-specific format and converting this data into the format required by a general tool such as *lexc* has merit for a wide variety of languages. Additionally, languages with complex phonologically conditioned allomorphy may benefit from a lexical treatment similar to the one described here, where an overgenerating lexicon with tagged allomorphs is filtered through a set of rules enforcing the conditions associated with those allomorphs.

Acknowledgements

This work was supported by the US National Science Foundation, Award 0534217. We also gratefully acknowledge Ida Mayer, J. Eliot DeGolia, Sai Venkateswaran, Paul Lundbland, and Shinjae Yoo, who prepared the corpus of Iñupiaq texts used to test the transducer, and three anonymous reviewers for their valuable feedback.

Abbreviations used

1	1st person
2	2nd person
3	3rd person
3R	3rd person reflexive
DU	dual
IND	indicative
INT	interrogative
OBJ	direct object
PL	plural
PRS	present
PST	past
SG	singular

8. References

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI, Stanford, California.

Kenneth R. Beesley. 2003. Finite-state morphological analysis and generation for Aymara. In *Proceedings of the Workshop on Finite State Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 10), April 13–14 2003, Budapest, Hungary*, pages 19–26.

Kenneth R. Beesley. 2004a. Downtranslation of XML dictionaries to lexc LEXICONs: Third draft, October 12. Published online: `http://www.stanford.edu/~laurik/fsmbook/clarifications/xmldowntrans.html`.

Kenneth R. Beesley. 2004b. Morphological analysis and generation: A first step in natural language processing. In Julie Carson-Berndsen, editor, *First Steps in Language Documentation for Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, Proceedings of the SALTMIL Workshop at LREC 2004, May 24, Lisbon, Portugal*, pages 1–8.

Lawrence D. Kaplan. 1981. *Phonological Issues in North Alaskan Inupiaq*. Number 6 in Research Papers. Alaska Native Language Center, Fairbanks, Alaska. Published version of Kaplan's 1979 doctoral dissertation.

Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *COLING 1992, 14th International Conference on Computational Linguistics, August 23–28, Nantes, France*, pages 141–148.

András Kornai. 1999. Extended finite state models of language. In Adrás Kornai, editor, *Extended Finite State Models of Language*, Studies in Natural Language Processing, pages 1–5. Cambridge University Press, Cambridge, England and New York, New York.

Michael E. Krauss. 2007. Native languages of Alaska. In Osahito Miaoka, Osamu Sakiyama, and Michael E. Krauss, editors, *The Vanishing Voices of the Pacific Rim*. Oxford University Press, Oxford, England.

Edna Ahgeak MacLean. 1981. *Iñupiallu Tanŋiḷḷu Uqaluŋisa Iḷaŋich = Abridged Iñupiaq and English Dictionary*. Alaska Native Language Center, Fairbanks, Alaska.

Edna Ahgeak MacLean. 1986a. *North Slope Iñupiaq Grammar: First Year*. Alaska Native Language Center, Fairbanks, Alaska, third edition.

Edna Ahgeak MacLean. 1986b. *North Slope Iñupiaq Grammar: Second Year (Preliminary Edition for Student Use Only)*. Alaska Native Language Center, Fairbanks, Alaska.

Edna Ahgeak MacLean. unpublished. North Slope Iñupiaq grammar: Third year. Draft manuscript.

Elaine Uí Dhonnchadha. 2003. Finite-state morphology and Irish. In *Proceedings of the Workshop on Finite State Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 10), April 13–14 2003, Budapest, Hungary*, pages 43–49.

Donald H. Webster and Wilfried Zibell. 1970. *Iñupiat Eskimo Dictionary*. Summer Institute of Linguistics, Fairbanks, Alaska. Digitized by Alaskool.org: `http://www.alaskool.org/language/dictionaries/inupiaq/default.htm`.

Shinjae Yoo. 2008. A smart OCR for Inupiaq. Final presentation of graduate-level semester project in language and information technologies at Carnegie Mellon University, December 12.