Finite-State Morphological Analysis and Generation for Aymara

Kenneth R. Beesley
Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan
France
ken.beesley@xrce.xerox.com

Abstract

A prototype morphological analyzer/generator for Aymara has been written using Xerox finite-state technology, and the full system is available for testing on the Internet. We describe the history of the project, the current implementation, and the plans for the future.

1 Introduction

Aymara is a highly agglutinating language spoken in Bolivia and Peru by roughly two million people. We[1] have written a morphological analyzer that accepts inflected orthographical Aymara words as input and returns analyses that separate and identify the constituent root and suffixes. To perform generation, i.e. the mapping from analyses to inflected words, the system is literally run in reverse.

2 History

In late 1988 and early 1989, Stuart Newton and I wrote a Two-Level (Koskenniemi, 1983) analyzer for Aymara (Beesley and Newton, 1989) as an exercise, basing it on Newton's knowledge of the language and Ross's *Rudimentos* (Ross, 1963). In two or three weeks we succeeded in creating a small prototype analyzer that worked for a number of non-trivial examples, but the system was never properly tested or expanded.

In late 1999, I resurrected my old notes and recreated this prototype using Xerox Finite-State Technology (Beesley and Karttunen, 2000). The project was again undertaken as an exercise, this time in exploring the most perspicuous ways to notate morphotactic facts. The inadequacies of the 1989 system soon became apparent after new testing and the acquisition of a more modern grammar (Hardman et al., 1988). The system was then completely rewritten, and a working demo was first made available on the Internet 17 April 2000.[2]

3 Implementation

The current prototype is based on XML dictionaries, a morphotactic grammar, and Replace Rules (Karttunen, 1995) that describe morphophonological alternations between underlying morphophonemic strings and surface strings written in the standard Alfabeto Unico orthography.

3.1 Dictionaries

3.1.1 Root Dictionary

The prototype XML dictionary of Aymara roots contains about 360 entries, most of them kindly supplied by Jorge Pedraza Arpasi, who maintains the Aymara Uta webpages.[3] A typical entry, for *uta*, is shown in Figure 1.

3.1.2 Suffix Dictionaries

The suffix classes of Aymara are also stored in XML dictionaries, including encodings that specify the morphophonology for each entry. In particular, suffixes must be marked to show how they affect the preceding vowel, either deleting it, lengthening it, or leaving it alone. Vowels at the end of suffixes must be marked as strong (resisting deletion), weak (deleting if any suffix follows), or neutral (being deleted, lengthened or left alone depending on the suffix that follows). The XML suffix lexicon corresponding

[1] The current system was written by Kenneth R. Beesley with abundant consultation from Stuart N. Newton, who is an Aymara speaker. I wish also to thank Jorge Pedraza Arpasi, Juan de Dios Yapita, Sabine Dedenbach-Salazar Sáenz, and Martha Hardman for responding to email queries from a total stranger.

[2] http://www.xrce.xerox.com/research/mltt/aymara

[3] http://www.aymara.org

```
<entry>
  <form>
    <lexical>uta</lexical>
  </form>
  <subentry cat="ncommon">
    <glosses>
      <english>
        <gloss>house</gloss>
      </english>
      <spanish>
        <glosa>casa</glosa>
      </spanish>
    </glosses>
  </subentry>
</entry>
```

Figure 1: The XML root entry for *uta*

to the verbal class I, subclass 5 of Hardman et al. (1988)[p. 97] is shown in Figure 2.

3.2 Morphotactics

The current morphotactic grammar is a formalization of the descriptions in Hardman et al. (1988). I have recently acquired a copy of Briggs (1993) and will soon begin mining it for additions and clarifications. Without implying any criticism, for I am highly indebted to the work done by such field linguists, it must be noted that making the description precise enough for a computational implementation requires some reading between the lines and independent analysis of the available examples. A graph of the currently implemented morphotactics can be downloaded from the website.

3.3 Alternations

Capturing the morphophonological facts, i.e. identifying the alternations between the abstract morphophonemic strings and the surface orthographical strings, and specifying them precisely as Replace Rules, was another challenge. To cite one of the simpler examples, the rule in Figure 3 deletes vowels that occur just before a suffix that "requires a previous consonant"; internally, such suffixes are marked with a preceding *ˆC* coding, and the 0 (zero) denotes the empty or zero-length string.

Similarly, Figure 4 shows the rule that lengthens vowels before suffixes like *:* 1>3-Future or *:ta* 2>3-Future that consist of or begin with

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE suffdict SYSTEM "suffdict.dtd">

<suffdict name="vI5">

<entry>
  <lex left="c">nuqa</lex>
  <glosses>
  <comment> HVY:101 </comment>
    <english>
      <gloss>setting down</gloss>
    </english>
    <spanish>
      <glosa>colocador</glosa>
    </spanish>
  </glosses>
</entry>

<entry>
  <lex left="v">qa</lex>
  <comment> has occurred after
       -ja vI2 </comment>
  <glosses>
    <english>
      <gloss>downward</gloss>
    </english>
    <spanish>
      <glosa>hacia abajo</glosa>
    </spanish>
  </glosses>
</entry>

<entry>
  <lex left="c">xata</lex>
  <glosses>
    <english>
      <gloss>above</gloss>
    </english>
    <spanish>
      <glosa>encima de</glosa>
    </spanish>
  </glosses>
</entry>

</suffdict>
```

Figure 2: XML dictionary for the class I subclass 5 verbal suffixes

```
[a | i | u] -> 0 || _ ^C
```

Figure 3: The rule that deletes vowels before a suffix that "requires a previous consonant"

```
a -> ä , i -> ï , u -> ü || _ :
```

Figure 4: Rule that lengthens vowels

the lengthening morphophoneme. Such Replace Rules are comparable to the rewrite rules used traditionally in phonological derivations (Chomsky and Halle, 1968). Other rules handle optional shortening of lengthened vowels, assimilations, and even a couple of cases of metathesis. The full set of alternation rules used in the current system can also be downloaded from the website.

3.4 Compilation into Finite-State Transducers

In the Xerox approach, the morphotactic description, including the dictionaries, and the morphophonological Replace Rules are compiled into data structures known as FINITE-STATE TRANSDUCERS or FSTs. These transducers are subsequently combined together in an operation known as composition to form a single LEXICAL TRANSDUCER (see Figure 5) that directly encodes the relation between analysis strings and surface orthographical strings.

When the lexical transducer is applied "in an upward direction" to a surface string like *utaxanaka*, the system returns the related string *uta [ncommon] +xa[1P_possessive] +naka[plural]*. And conversely, if the same transducer is applied "in a downward direction" to the analysis string *uta [ncommon] +xa[1P_possessive] +naka[plural]*, the system returns the related surface string *utaxanaka*. Transducers are inherently bidirectional.

Finite-state transducers are not computer programs or algorithms in the usual sense; rather they are data structures that encode relations and can be applied to input in either direction by pre-existing, language-independent algorithms. Xerox has built finite-state transducers to do morphological analysis for English, French, Spanish, Portuguese, Italian, Finnish, Hungarian, Arabic, etc., and the exact same application algorithms are used for all of them.

It is not appropriate or possible to go into an explanation of finite-state transducers in this report; suffice it to say that computing with finite-state transducers is mathematically elegant, resulting in computational systems that are flexible, compact, and efficient. Xerox finite-state morphological analyzers typically process thousands of words per second on modern workstations and high-end PCs, making it completely feasible to test them on corpora of millions of words.

4 Website

Finally, the working system was provided with an HTML interface and made publicly available on the Internet for testing and, hopefully, feedback from Aymarists. Input options include the Alfabeto Unico and the trivially different Yapita orthography, and glosses for the roots can be displayed in either Spanish or English. The current dictionary of roots is very small but can, of course, be expanded with new entries. Users are encouraged to contribute new root entries, and an HTML FORM is provided to facilitate submissions. The full root dictionary is downloadable and will be kept in the public domain to encourage contributions.

5 The Future

Much work remains to be done. In addition to serious dictionary work, which can take years, there is a need to collect reliable online corpora and use them continuously in testing. Known fuzzy areas in the description of Aymara morphology, such as the treatment of the 'complemento cero' and the dropping of word-final vowels in general, need clarification from the field.

6 Conclusions

The present project was an exercise, by a non-Aymara-speaking computational linguist, to explore the most perspicuous way to notate natural-language morphotactics in general; this work will lead to the improvement of the programming tools themselves. It was also an exercise in computerizing the semi-formal descriptions of field linguists to build a working system. Just as this project benefited from, indeed depended on, the work of the field linguists, it is expected that the discipline of writing computational grammars will result in insights and

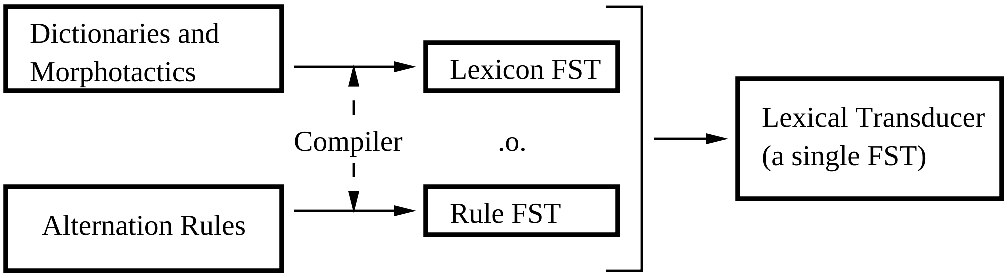

Figure 5: Creation of a Lexical Transducer

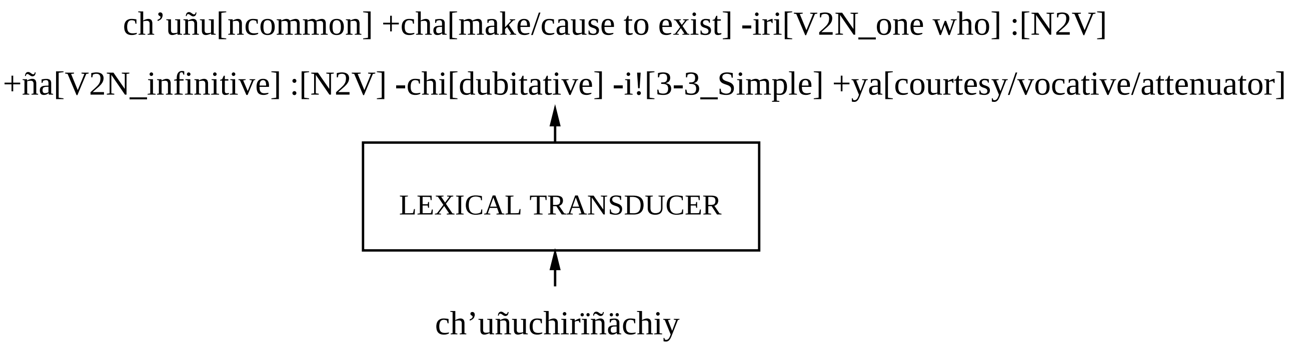

Figure 6: One analysis of ch’uñuchirïñächiy

corrections that flow back to improve the more traditional “paper grammars”. Field linguists and computational linguists have everything to gain by working together.

One result of this work has been a clarification of the alternation between the *ya* and *:* allomorphs used for verbalization (Beesley, 2000). This alternation, presented as a “complejidad” requiring future study in Briggs (1993), created serious practical problems for the morphological analyzer until the phenomenon was understood and the rules were formalized. In this, as in many other cases, computational linguistics forces you to face up to the errors and gaps in your descriptions.

References

Kenneth R. Beesley and Lauri Karttunen. 2000. *Finite-State Morphology: Xerox Tools and Techniques*. Submitted to Cambridge University Press.

Kenneth R. Beesley and Stuart N. Newton. 1989. Computer analysis of Aymará morphology: A two-level, finite-state approach. In *Proceedings of the Fifteenth Annual Deseret Language and Linguistics Symposium*, pages 126–144, Provo, Utah, March 13–14. Brigham Young University.

Kenneth R. Beesley. 2000. A note on phonologically conditioned selection of verbalization suffixes in Aymara. Technical report, Xerox Research Centre Europe, July.

Lucy Therina Briggs. 1993. *El Idioma Aymara: Variantes regionales y sociales*. ILCA, La Paz.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper and Row, New York.

Martha J. Hardman, Juana Vásquez, and Juan de Dios Yapita. 1988. *Aymara: Compendio de estructura fonológica y gramatical*. ILCA, La Paz.

Lauri Karttunen. 1995. The replace operator. In *ACL’95*, Cambridge, MA. cmp-lg/9504032.

Kimmo Koskenniemi. 1983. Two-level morphology: A general computational model for word-form recognition and production. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Ellen H. Ross. 1963. *Rudimentos de gramática aymara*. Misión Bautista Canadiense, La Paz.