Proceedings of the SALTMIL Workshop at LREC 2004

First Steps in Language Documentation for Minority Languages

Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation

SALTMIL
http://isl.ntf.uni-lj.si/SALTMIL/

Preface

SALTMIL (http://isl.ntf.uni-lj.si/SALTMIL/) is the ISCA Special Interest Group focusing on *S*peech *A*nd *L*anguage *T*echnology for *MI*nority *L*anguages. Minority or "lesser used" languages of the world are under increasing pressure from the major languages (English, in particular), and many of them lack full political recognition. While some minority languages have been well researched linguistically, most have not, and the vast majority do not yet possess basic speech and language resources (such as text and speech corpora) that are sufficient to permit research or commercial development of products. If this situation were to continue, the minority languages would fall a long way behind the major languages, as regards the availability of commercial speech and language products. This in turn will accelerate the decline of those languages that are already struggling to survive. To break this vicious circle, it is important to encourage the development of basic language resources as a first step. In order to address this issue, SALTMIL organises workshops with the aim of bringing researchers together and fostering collaboration in this important area. Workshops have been held in conjunction with the past three LREC conferences.

1. *Language Resources for European Minority Languages* (LREC1998) Granada, Spain.
2. *Developing Language Resources for Minority Languages: Re-usability and Strategic Priorities* (LREC2000) Athens, Greece.
3. *Portability Issues in Human Language Technologies* (LREC2002) Las Palmas de Gran Canaria, Spain.

This volume is the proceedings of the fourth SALTMIL Workshop held at LREC 2004. The theme of the workshop is *First Steps in Language Documentation for Minority Languages*. Following previous formats, the workshop will begin with a series of invited oral presentation and end with a poster session. The first session consists of invited papers (Beesley, Artola-Zubillaga and Gibbon) addressing computational linguistic tools for morphology, lexicon and corpus compilation, followed by a presentation on collaboration opportunities and sources of funding within the EU Sixth Framework (Petek). The programme committee were delighted at the response to the call for posters. The committee reviewed 31 submissions for this session from researchers working with minority languages. 17 contributions were selected for presentation at the workshop. All papers are published in these proceedings.

Julie Carson-Berndsen, *Proceedings Editor*
April 2004

The Workshop Programme

Oral Session: First Steps for Language Documentation

08:00 Welcome and Introduction. Bojan Petek (University of Ljubljana)
08:15 "Morphological Analysis and Generation: A First Step in Natural Language Processing" Kenneth R. Beesley (Xerox Research Centre Europe)
08:45 "Laying Lexical Foundations for NLP: the Case of Basque at the Ixa Research Groups" Xabier Artola-Zubillaga (University of The Basque Country)
09:15 "First steps in corpus building for linguistics and technology" Dafydd Gibbon (University of Bielefeld)

09:45 Coffee Break

Oral Session: HLT and the Coverage of Languages

10:15 "First steps in FP6" Bojan Petek (Univ. of Ljubljana)
10:45 Panel discussion - Invited speakers will be included as panellist members.

Poster Session

12:00 – 13:30

1. Atelach Alemu, Lars Asker & Gunnar Eriksson: Building an Amharic Lexicon from Parallel Texts
2. Jens Allwood & A. P. Hendrikse: Spoken Language Corpora in South Africa
3. Emily Bender Dan Flickinger, Jeff Good, Ivan Sag: Montage: Leveraging advances in grammar engineering, linguistic ontologies, and mark-up for the documentation of underdescribed languages.
4. Akshar Bharati, Rajeev Sangal, Dipti M Sharma, Radhika Mamidi: Generic Morphological Analysis Shell
5. Bordel G., Ezeiza A. , Lopez de Ipina K., Méndez M., Peñagarikano M., Rico T., Tovar C., Zulueta E.: Linguistic Resources and Tools for Automatic Speech Recognition in Basque.
6. Montserrat Civit, Núria Bufí and Ma. Pilar Valverde: Building Cat3LB: a Treebank for Catalan.
7. Mike Maxwell: From Legacy Lexicon to Archivable Resource.
8. László Németh, Viktor Trón, Péter Halácsy, András Kornai, András Rung , István Szakadát: Leveraging the open source ispell codebase for minority language analysis.
9. Ailbhe Ní Chasaide, Martha Dalton, Mika Ito, Christer Gobl: Analysing Irish prosody: A dual linguistic/quantitative approach.
10. Attila Novák: Creating a Morphological Analyzer and Generator for the Komi language.
11. Delyth Prys, Briony Williams, Bill Hicks, Dewi Jones, Ailbhe Ní Chasaide, Christer Gobl, Julie Carson-Berndsen, Fred Cummins, Máire Ní Chiosáin, John McKenna, Rónán Scaife, Elaine Uí Dhonnchadha: WISPR: Speech Processing Resources for Welsh and Irish.
12. Rajmund Piotrowski, Yuri Romanov: Imperial and Minority Languages in the Former USSR and in the Post-Soviet Area
13. Ksenia Shalonova and Roger Tucker: Issues in Porting TTS to Minority Languages.
14. Kiril Simov, Petya Osenova, Alexander Simov, Krasimira Ivanova, Ilko Grigorov, Hristo Ganev: Creation of a Tagged Corpus for Less-Processed Languages with ClaRK System.
15. Oliver Streiter, Mathias Stuflesser, Isabella Ties: CLE, an aligned Tri-lingual Ladin-Italian-German Corpus. Corpus Design and Interface.
16. Nicholas Thieberger: Building an interactive corpus of field recordings.
17. Trond Trosterud: Porting morphological analysis and disambiguation to new languages.

Workshop Organisers/Programme Committee

Atelach Alemu	Addis Ababa University, Ethiopia
Julie Carson-Berndsen	University College Dublin, Ireland
Bojan Petek	University of Ljubljana, Slovenia
Kepa Sarasola	University of the Basque Country, Donostia
Oliver Streiter	EURAC; European Academy, Bolzano/Bozen, Italy

Table of Contents

Author Index

Morphological Analysis and Generation: A First-Step in Natural Language Processing

Kenneth R. Beesley

Xerox Research Centre Europe
6, chemin de Maupertuis
38240 MEYLAN, France
Ken.Beesley@xrce.xerox.com

Abstract

Computer programs that perform morphological analysis and generation are a useful bridge between language resources, such as corpora, lexicons and printed grammars, and the overall field of natural language processing, which includes tokenization, spelling checking, spelling correction, non-trivial dictionary lookup, language teaching and comprehension assistance, part-of-speech disambiguation, syntactic parsing, text-to-speech, speech recognition, and many other applications. This paper is an overview of morphological analysis/generation using finite-state techniques, listing available software, showing how existing language resources can be used in building and testing morphology systems, and explaining how root-guessing morphological analyzers can help expand those resources by actively suggesting new roots that need to be added to the lexicon.

1. Introduction

Creating an automatic morphological analyzer/generator is just one step in starting natural language processing for any language; but especially for minority, emerging or generally lesser-studied languages, it is often a practical and extremely valuable first step, making use of corpora, lexicons, morphological grammars and phonological rules already produced by field linguists and descriptive linguists. If the linguistic knowledge and lexical resources are sound, and if the data can be formatted in precise ways, there are a number of readily available software packages that can take the static data and compile them into active computer programs that are interesting in themselves and which are necessary components in larger natural-language applications.

Building a morphological analyzer and testing it on real text is a healthy exercise for correcting and completing morphological descriptions, rules and lexicons; to encode a morphological analyzer that really works, without overgeneration or undergeneration, you often have to think harder, and more precisely, about the data than you ever have before. Modern morphological analyzers can be applied, in a matter of minutes, to a corpus of a million words, testing the accuracy and completeness of the model to an extent that would never be practical by hand. It is also possible to define "root-guessing" morphological analyzers that actively suggest new roots that need to be added to the lexicons.

This paper will proceed with a broad overview of morphological analysis and generation and, in particular, explain how such programs can be implemented using finite-state techniques (Roche and Schabes, 1997; Antworth, 1990; Beesley and Karttunen, 2003). The translation of existing linguistic resources, such as XML dictionaries, into formats suitable for finite-state compilers will also be discussed, as will the range of finite-state implementations now available. Finally, a selective list of current morphology projects will be mentioned, showing that finite-state techniques can be and are being applied to languages all over the world.

2. Overview of Morphological Analysis and Generation

2.1. The Study and Description of Morphology

Morphology is the branch of linguistics that studies words. In a tradition going back as far as Panini (520BC?–460BC?) who wrote a grammar of Sanskrit,[1] morphology has two subgoals

- To describe the MORPHOTACTICS, the grammar for constructing well-formed words out of parts called MORPHEMES, and
- To describe the MORPHOPHONOLOGY, the rules governing phonological and orthographical alternations between underlying forms and surface spoken or written forms.

In what follows, I assume that the input to a morphological analyzer is digitized text in a standard orthography, but the techniques also apply to phonology and indeed started in phonology (Karttunen, 1991).

2.2. Morphotactics

It is not possible in this overview to go into all the complications of natural-language morphotactics, which can include compounding, circumfixation, infixation, partial and complete reduplication, and interdigitation of morphemes. For present purposes we will restrict our examples to straightforward word-formation using prefixes and suffixes, which can be challenging enough in many cases.

Let us assume a prototypical natural language, in which verbs are built on a verb root that may allow, or require, certain affixes (prefixes and/or suffixes) that are drawn from closed, very finite, sets. Figure 1 is a skeleton diagram

[1] The dates given for Panini are only guesses. In any case, his Astadhyayi, a formal description of Sanskrit morphology and "sandhi" alternations, continues to impress modern linguists and computer scientists. `http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Panini.html`

Figure 1: Simple morphotactic construction can often be visualized as one or more prefixes, drawn from very finite classes, followed by a root, followed by one or more suffixes, also drawn from very finite classes. The root class is typically the only open class, and the morphemes must typically appear in a strict relative order.

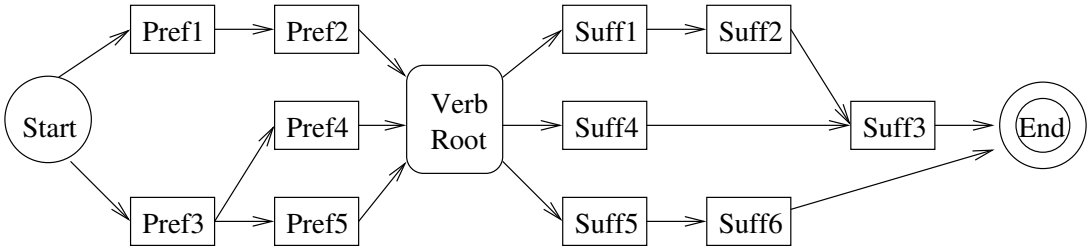


Figure 2: More complicated morphotactic descriptions are more obviously graph-like, with a start node, perhaps dozens of morpheme classes, and one or more end nodes. Each path through the graph, from the start state to a final state, represents a morphotactically well-formed, but still phonologically or orthographically abstract, word.

of verb construction for such a straightforward language, where each box represents a set of morphemes. A morphotactically well-formed verb is constructed by selecting one morpheme from each set—or perhaps none, if the affix is optional—and concatenating them together in the strict order implied by the diagram. Nouns and other categories may be built on a similar scheme, though usually with distinct sets of roots and affixes.

The facts of morphotactics can easily become more complicated, requiring descriptions that are more obviously graph-like, with a start state and one or more final states, as in Figure 2. A morphotactically well-formed word is constructed by starting at the start state and finding some path through the directed graph to a final state, accumulating a morpheme from each class encountered on the path (or perhaps none, if the class is optional).

For finite-state computational morphological analysis and generation, the linguist must define such morphotactic networks with great precision, either in direct graphic form or using textual grammars that compile into such networks, depending on the software being used.

2.2.1. Morphophonological Alternations

In real natural languages, the straightforward concatenation of morphemes seldom yields a finished word but rather an abstract or "underlying" word that is subject to phonological and/or orthographical alternations. Except in artificially regular languages like Esperanto, funny things happen when morphemes are strung together, especially at morpheme boundaries. The alternations between underlying strings and surface strings can involve assimilation, deletion, lengthening, shortening and even dissimilation, epenthesis, metathesis, etc. Such alternations are traditionally described in "rewrite rules" that specify an input, an output, and right and left contexts which must match for the rule to apply.

A natural language may easily require dozens of rules to describe the morphophonological alternations, and such rules were written with great care by Panini and many other linguists[2] long before computers were available. With modern software to support morphological analysis, rules similar to the following can be easily written and tested, though the exact syntax of the rules and their expressive flexibility differ among the various implementations.

Deletion	t $\rightarrow \epsilon$ / f _ e n
Epenthesis	$\epsilon \rightarrow$ p / m _ k
Assimilation	s $\rightarrow$ z / Vowel _ Vowel
Metathesis	s k $\rightarrow$ k s / _ #

Table 1: Most implementations of finite-state morphology allow linguists to express phonological and orthographical alternations using high-level rewrite rules.

2.2.2. Black Box Morphological Analysis

In the most theory-neutral way, a morphological analyzer can be viewed as a black box as shown in Figure 3. Analysis, typically visualized as an "upward" process, is performed by feeding a word into the bottom of the black box, which somehow undoes the morphotactic processes and morphophonological alternations, and outputs zero or more analyses: zero analyses would indicate that the word was not successfully analyzed, and more than one analysis indicates ambiguity. What the analysis looks like is very dependent on theory and implementation, so it's hard to generalize; but in principle a morphological analyzer should separate and identify the root and other morphemes.

Other desiderata for a morphological analyzer are speed, robustness, and scalability to handle all the words of a language. And it would be extremely attractive if the very same black box could run in the generation direction as

[2] One of my favorite examples is a set of 60-odd rewrite rules written to map underlying Mongolian strings into the standard Cyrillic orthography (Street, 1963).

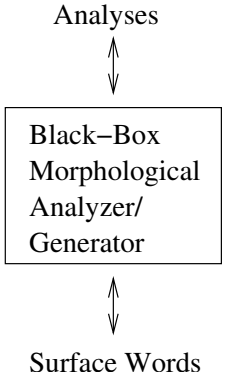


Figure 3: A morphological analyzer can be visualized as a black-box that accepts a surface word and returns zero or more analyses. Ideally the black box is bidirectional, also capable of accepting an analysis and returning zero or more surface words.

well, allowing you to input an analysis (whatever that looks like) in the top and to get out zero or more surface forms corresponding to the analysis. A black box component that performs morphological analysis and generation could be implemented any number of ways. Twenty years of experience, with natural-language projects all over the world, have shown that such morphological analyzer/generators can be implemented elegantly as FINITE-STATE TRANSDUCERS.

It is not appropriate here to go into a formal description of finite-state automata; this information is easily available elsewhere (Roche and Schabes, 1997; Beesley and Karttunen, 2003). For present purposes, we can state that finite-state implementations of natural-language applications, compared to the alternatives, are generally smaller, faster, and more scalable and maintainable. Finite-state transducers are bidirectional. Finite-state methods certainly cannot do everything in natural language processing; but where they are appropriate, they are extremely attractive.

The overall claim of finite-state morphology, at least in the Xerox implementation that I know best, is that both morphotactic descriptions and alternation rules can be compiled into finite-state transducers, as shown in Figure 4. The application of the rules, compiled into a rule transducer, to the abstract words, compiled into a lexicon transducer, is accomplished by the finite-state operation of COMPOSITION, which can be simulated in runtime code or, in some implementations, performed once and for all at compile time, yielding a single LEXICAL TRANSDUCER (Karttunen et al., 1992) that serves as the morphological black box.

3. Available Software for Finite-State Morphological Analysis

3.1. Two-Level Morphology

Writing a morphological analyzer/generator is a kind of computer programming, but finite-state programming is declarative rather than algorithmic. There are in fact a number of implementations of finite-state morphology available, and I will go through those implementations that I know to be mature and available.

The first practical implementation was by Kimmo Koskenniemi, now a professor of Linguistics at the University of Helsinki, who had been exposed to finite-state theory at the Xerox Palo Alto Research Center. Koskenniemi's "Two-Level Morphology", presented in his 1983 thesis (Koskenniemi, 1983; Koskenniemi, 1984) was popularized by Lauri Karttunen at the University of Texas in a re-implementation called KIMMO (Karttunen, 1983) and was later made easily available to all in a well-documented implementation called PC-KIMMO from the Summer Institute of Linguistics (SIL) (Antworth, 1990; Sproat, 1992).[3]

Two-Level Morphology provides a syntax to define morphotactics and morphophonological alternations, but it doesn't really have an underlying library of finite-state algorithms. There was no automatic rule compiler, which required developers to hand-compile their rules into finite-state transducers, which is both tricky and tedious. Later Koskenniemi collaborated with Xerox researchers to build an automatic rule compiler (Koskenniemi, 1986; Karttunen et al., 1987) and an independent compiler called KGEN, which compiles simple two-level-style rules, eventually became available.[4]

Two-Level Morphology, in one form or another, has been used by a whole generation of students in computational linguistics and has been applied, academically or commercially, to languages all around the world, including Finnish, Swedish, Norwegian, Danish, German, Russian, Swahili, English,[5] Hungarian, Mongolian, Akkadian (Kataja and Koskenniemi, 1988), Looshutseed and other Salishan languages (Lonsdale, 2003), Tagalog (Antworth, 1990), etc. I myself sometimes admit to having used PC-KIMMO to write a morphological analyzer for Klingon (Beesley, 1992b; Beesley, 1992a), an agglutinating language that was invented for the Star Trek series (Okrand, 1985).

3.2. Xerox Finite-State Tools

The proof that phonological and morphological alternation rules, as used by linguists, were only finite-state in power, and could be implemented as finite-state transducers, was presented in 1972 by C. Douglas Johnson, whose book (Johnson, 1972) was unfortunately overlooked; the same insight was rediscovered later by Xerox researchers (Kaplan and Kay, 1981; Karttunen et al., 1992; Kaplan and Kay, 1994), but the implementation of the finite-state theory in practical software libraries and automatic compilers has proved to be surprisingly difficult. Xerox researchers have been working on it for over twenty years.

I work for Xerox, and so I'm natural best acquainted with this implementation.[6] I have used the Xerox finite-state software to help write lexical transducers for Spanish, Portuguese, Italian, Dutch and Arabic, plus significant

[3] http://www.sil.org/pckimmo/

[4] http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/morph/pc_kimmo/kgen/0.html, http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrinfo?KGEN

[5] http://www.lingsoft.fi/demos.html

[6] http://www.xrce.xerox.com/competencies/content-analysis/fst/

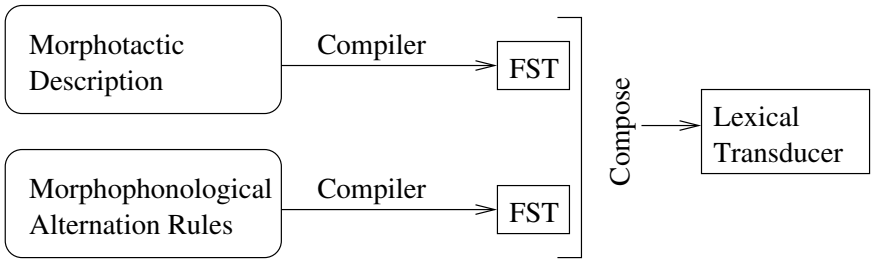


Figure 4: The general claim of finite-state morphology, at least in the Xerox tradition, is that both morphotactic grammars and alternation rules can be compiled into finite state transducers (FSTs). The application of the rule FST to the morphotactic (lexicon) FST is performed by the operation of COMPOSITION, producing a LEXICAL TRANSDUCER, a single FST that functions as an analyzer/generator.

prototypes for Malay and Aymara, a language spoken in Bolivia, Peru and Chile. With Lauri Karttunen, I've written a textbook called *Finite State Morphology* (Beesley and Karttunen, 2003) that shows how to write Xerox-style morphological analyzer/generators, and related lexical applications; the book includes a CD-ROM with the code, compiled for Solaris, Linux, Mac OS X and Windows, with a non-commercial license.[7]

The Xerox code includes **xfst**, an interface that provides access to the individual algorithms of the finite-state library (union, subtraction, complementation, concatenation, intersection, cross-product, composition, etc.), symbol-table manipulation, file input-output, and compilers that turn wordlists and regular-expressions into finite-state automata. Most computer programmers are already familiar with regular expressions, and, theoretically, any finite-state automaton, including those that perform morphological analysis and generation, can be defined using just regular expressions. In practice, however, regular expressions are not always convenient or intuitive, especially for describing morphotactics, and Xerox therefore provides another high-level language and associated compiler called **lexc**, which lets you define finite-state automata using right-recursive phrase-structure grammars.[8]

3.3. INTEX/UNITEX

The INTEX system of Max Silberztein[9] represents a rather different finite-state tradition, based on work from the University of Paris VII. INTEX users construct finite-state networks directly using a graphical user interface. A Unicode-capable clone called UNITEX is also available.[10]

3.4. Fsa Utils 6

Gertjan van Noord and Dale Gerdemann of the University of Groningen have produced a Prolog-based implementation of finite-state theory called Fsa Utils 6. The code, including sources, and abundant documentation is available from their website.[11]

3.5. AT&T FSM Library and Lextools

One of the main commercial competitors to the Xerox Finite State Tools is AT&T's FSM Library[12] and their set of programming formats called Lextools,[13] which include compilers for morphotactic descriptions, replace rules, regular expressions and inflectional paradigms. AT&T has used these tools extensively for speech recognition and text-to-speech systems.

3.6. Other Implementations

Other implementations that I admittedly know less well include Grail,[14] Jan Daciuk's Utils,[15] and the work of Tomasz Kowaltowsky.[16] All the implementations listed are based on the same mathematical foundations and share a certain family resemblance. In some cases, source files written for one implementation can be converted easily to the syntax of another. A user interested in finite-state development must do a bit of homework to select an implementation that fits his or her needs and tastes.

4. XML and Morphology

4.1. Use of Corpora and Lexicons in XML

In a conference dedicated to linguistic resources, it is proper to spend a minute discussing how existing resources can be used in building and testing morphological analyzer/generators. The resources that are most obviously applicable are corpora and lexicons, which are now almost universally stored in electronic form, and increasingly in XML. Corpora are obviously and directly useful for testing, and XML dictionaries can often be converted trivially into source code for the various software implementations.

In the past, before XML was available, Xerox developers typically wrote dictionaries directly in **lexc** format,

[7] http://www.fsmbook.com

[8] General phrase-structure grammars can define languages and relations that go beyond regular power, and so could not be implemented as finite-state automata, but if the grammars are constrained to be right-recursive (and/or left-recursive) then, like **lexc**, they are restricted to regular power.

[9] http://www.nyu.edu/pages/linguistics/intex/

[10] http://www-igm.univ-mlv.fr/~unitex/

[11] http://odur.let.rug.nl/~vannoord/Fsa/

[12] http://www.research.att.com/sw/tools/fsm/

[13] http://www.research.att.com/sw/tools/lextools/

[14] http://www.csd.uwo.ca/research/grail/

[15] http://www.eti.pg.gda.pl/~jandac/fsa.html

[16] http://colibri.ic.unicamp.br/~tomasz/

which in retrospect was kind of a dead end. The dictionaries required for morphological analysis and generation are typically sparse, and very hard to share with other projects. Today XML is a widely available and justifiable popular framework for "marking-up" almost any kind of data that has hierarchical structure, with dictionaries being a perfect application. XML is in fact not a single markup-language, but a framework in which we can define an infinite number of custom markup-languages for specific purposes. XML files themselves are typically quite simple, consisting of an XML declaration at the top and a single top-level "element", beginning with a "start tag" and terminated with a matching "end tag". The top-level element contains other elements, text, and mixtures of elements and text, organized in a hierarchical tree structure, as in the following sketch of an XML dictionary for Aymara.

```
<?xml version="1.0"?>

<dictionary>
  <head>
    <title>Ken's Aymara Dictionary</title>
    <date>2004-03-21</date>
    <version>1.0</version>
    <copyright>Copyright (c) 2004 Xerox
Corporation.  All rights reserved.
    </copyright>
    <comment>Should conform to Relax NG
Schema aymaradic.rng</comment>
  </head>
  <body>
    <entry>...</entry>
    <entry>...</entry>
    ...
  </body>
</dictionary>
```

Each type of XML markup language must be defined by a grammar that defines the names of the elements, the tree-like structure of the document, and other details. As shown in the sketch above, a typical XML dictionary might contain a `head` element, containing `title`, `date`, `version`, `copyright` information, and other elements containing meta-information, followed by a `body` element containing a list, perhaps huge, of `entry` elements.

The `entry` elements would have their own internal structure necessary to store dictionary-entry information for Aymara (or whatever). The following is a real example from my current Aymara dictionary.

```
  <entry>
    <form>
      <lexical>achu</lexical>
    </form>

    <subentry cat="ncommon">
      <comment>from J.P. Arpasi</comment>
      <glosses>
        <english>
          <gloss>fruit</gloss>
        </english>
        <spanish>
          <glosa>fruto</glosa>
        </spanish>
      </glosses>
    </subentry>

    <subentry cat="verb">
      <comment>from J.P. Arpasi</comment>
      <comment> see NVY:19, needs non-human
subject</comment>
      <glosses>
        <english>
          <gloss>produce</gloss>
          <gloss>ripen</gloss>
        </english>
        <spanish>
          <glosa>producir(se)</glosa>
          <glosa>madurar</glosa>
        </spanish>
      </glosses>
    </subentry>

  </entry>
```

A richer format might include fields for phonology, additional subcategory information, glosses for more languages, prose definitions, and pointers to synonyms and antonyms. Ideally your XML language should include fields for all the information needed to support multiple applications, with the understanding that any particular project will use only a subset.

The advantages of XML are many. It is a free and unencumbered standard, and most of the software for validating and processing XML files is completely free and even open-source. Your XML files are plain text (including Unicode), and in general you control your own data instead of being at the mercy of proprietary database formats.

4.2. Downtranslation of XML

An XML dictionary may be very rich, containing a wide variety of information needed to support a number of different projects. To write a morphological analyzer, you typically need to parse the XML dictionary, extract some subset of the information for each entry, and then write it out in a format suitable for a finite-state compiler such as Xerox's **lexc**, AT&T's Lextools, etc. And as long as the XML contains the necessary information, in some reasonable structure, a common XML file can even be downtranslated to multiple output formats. In 2004, using XML and XML translation is the best bet for making your dictionaries sharable, flexible, and immortal.

XML dictionary downtranslation is a kind of computer programming, and this usually requires some training or some help from computer programmers. But as XML files are effectively pre-parsed, with clearly labeled tree structure, such manipulations are typically very easy. Widely available standards for XML translation and downtranslation include XSLT,[17] which some like and others hate, event-driven SAX (Simple API for XML) parsing,[18] which is too low-level for some tastes, and DOM (the Document Object Model),[19] which is high-level, powerful and intuitive. The main problem with DOM parsers is that they

[17] `http://www.w3.org/TR/xslt`
[18] `http://www.saxproject.org/`
[19] `http://www.w3.org/DOM/`

read in whole XML files and store them as tree structures in memory, which can too easily cause memory problems, especially for large XML files for natural-language dictionaries and corpora. I have found Perl's XML::Twig module[20] to be an excellent compromise between SAX and DOM, allowing an XML document to be processed in entry-sized "chunks", which can be deleted when you are through with them. It is very hard to generalize here, because tastes and needs differ so much, but the main problem with XML processing these days is not finding a solution, but choosing among a bewildering variety of possible solutions.

5. Root-Guessing Morphological Analyzers

Finite-state morphological analyzers are ultimately based on dictionaries, and when a morphological analyzer is first written, the underlying dictionary is typically rather small, and the coverage of the analyzer is correspondingly low. Broad-coverage dictionaries for open-classes like noun roots and verb roots will require tens of thousands of entries and may take years to collect.

While testing early versions of your analyzer on real text, you can collect the failures, i.e. the words that aren't successfully analyzed, study them to figure out what is wrong, and progressively improve your system. At some point, if you've done your work well, improvement should reduce mostly to adding new roots to the dictionary. But using finite-state techniques, you can also build a modification of your analyzer that actively suggests new roots that need to be added to the dictionary. Such "guessers" or "root-guessing analyzers" can be applied when the strict analyzer, based on enumerated roots, fails.

Figure 5 shows a morphotactic network for nouns based on a lexicon of enumerated, attested roots; when you first begin your work, there may be only hundreds, or even just a few dozen entries in the noun-root sublexicon. Figure 6 shows the same system, but with the subnetwork of attested roots replaced by a network that accepts any phonologically possible root. The subnetwork that accepts phonologically possible roots can be defined using regular expressions, in any appropriate detail, and the resulting analyzer will accept any noun based on a phonologically possible root.

The runtime code that applies your analyzers to input words can be instructed to try the strict analyzer first, resorting to the guesser only when strict analysis fails. Analyses produced by the guesser will show roots that can easily be labeled as guesses, and these can be forwarded to your lexicographer for possible addition to the strict lexicon of enumerated roots. The definition and application of root-guessing analyzers using Xerox software is described in our book (Beesley and Karttunen, 2003, pp. 444–451), and the techniques are doubtless convertible to other implementations as well.

6. Morphology Projects

6.1. Interdisciplinary Cooperation

A practical morphology project requires linguistic, lexicographic, and computational expertise, seldom found all together in one person. Thus you often need collaboration among computer programmers who have a healthy interest in natural language, lexicographers, who manage corpora and dictionaries, and traditional field linguists or descriptive linguists, who know the language inside out, can tell when the output is right, and can work with informants where necessary.

I have worked in the area of computational morphology, and related lexical-level natural-language processing, for about 20 years. For the last 16 years, I've been using finite-state techniques, working directly on Spanish, Portuguese, Dutch, Italian, Malay, Aymara and especially Arabic,[21] but I don't speak all these languages. I'm perhaps best known for my work on Arabic, a language that I definitely do not speak and have never studied formally. Our Arabic system resulted from teamwork involving myself, as computational linguist, two Arabic linguist/lexicographers, Tim Buckwalter and Martine Pétrod, who had spent years of their lives studying Arabic and building paper dictionaries, with occasional help from native speakers. Such interdisciplinary teamwork is necessary and healthy, and it can also be attractive to project sponsors.

6.2. Some Commercial Finite-State Morphology Projects

Using various finite-state implementations, finite-state morphological analyzers have been written for the big, obviously commercial Indo-European languages, including English, French, German, Spanish, Italian, Portuguese, Dutch, Russian, Czech, Polish, Swedish, Norwegian and Danish. But in a refreshing break from tradition, most of the initial push for finite-state morphology came from Finns, Kimmo Koskenniemi and Lauri Karttunen, who needed a computational framework that would work well for their highly agglutinating non-Indo-European language. Similar work quickly extended to other non-Indo-European languages including Hungarian, Swahili, Japanese, Tagalog, Turkish, Akkadian and Arabic.

6.3. Some Current Projects for Lesser-Studied Languages

I've had great fun teaching and consulting with linguists working on a number of emerging or lesser-studied languages around the world, and what follows is a necessarily selective list of current projects that I'm aware of, most of which are using the Xerox software. The purpose here is to show that finite-state techniques can be and are being applied to languages all over the world, and perhaps to inspire a few more projects for lesser-studied languages.

In this workshop we will hear a report from the team at the University of the Basque Country, including Iñaki Alegria, Xabier Artola and Kepa Sarasola, who are using a variety of techniques, including finite-state morphology, to build natural language applications for Basque, another decidedly non-Indo-European language.[22] This project is notable for the innovations required to handle dialect variants

[20] http://www.xmltwig.com/xmltwig/

[21] http://www.arabic-morphology.com

[22] http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/a/Alegria:I=ntilde=aki.html

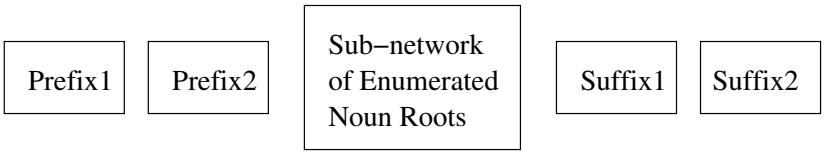


Figure 5: A strict morphological analyzer for nouns will analyze only those nouns what are based on the roots explicitly enumerated in the dictionary. When your dictionary of roots is small, as at the beginning of many projects, the coverage of the analyzer will be poor.

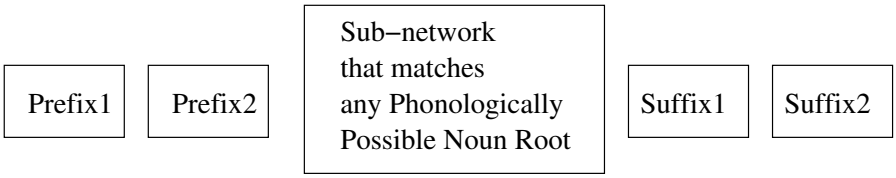


Figure 6: A root-guessing variant of your noun analyzer can be created by replacing the sub-network that matches only explicitly enumerated noun roots with a sub-network that matches any phonologically possible noun root.

of this still-emerging language, where the unified Batua dialect is not yet firmly established.

Na-Rae Han at the University of Pennsylvania has recently finished (as much as any dictionary-based project is ever finished) a huge system for Korean morphology.[23] Shuly Wintner[24] of the University of Haifa has been working for years on Hebrew and Arabic, along with students including Shlomo Yona[25] and Yael Cohen-Sygal.[26] Sisay Fissaha at the University of the Saarland has been working on Ahmaric morphology,[27] and Trond Trosterud of the University of Tromsø is working on four different dialects of Sámi,[28] once known as Lappish, and has plans to tackle Komi, Nenets, Udmurt and Mari. Elaine Uí Dhonnchadha of the Linguistics Institute of Ireland has written a morphological analyzer for Irish[29] and continues to expand the dictionary.

For American Indian languages, I'm aware of work by Jonathan Amith and Mike Maxwell, of the Linguistic Data Consortium, on Nahuatl, a Uto-Aztecan language. Bill Poser, also of LDC, has a project for Carrier, an Athapascan language. Deryle Lonsdale of Brigham Young University is working on several Salishan languages using PC-KIMMO (Lonsdale, 2003), and I myself have done some work on Aymara (Beesley, 2003).

I've now traveled down to South Africa twice, and will return again in September, to give a finite-state programming course to linguists wanting to start morphology projects for Bantu languages. And I'll be consulting with teams already working on Zulu (Sonja Bosch and Laurette Pretorius, University of South Africa), Xhosa (Jackie Jones, Kholisa Podile and Joseph Mfusi, University of South Africa; Duke Coulbanis, MSc student in Computer Science, University of South Africa), Ndebele (Axel Fleisch, U.C. Berkeley, University of Cologne), Northern Sotho (Albert Kotze, Lydia Mojapelo and Petro du Preez, University of South Africa; Winston Anderson, Byte Technology Group), and Setswana (Gerhard van Huyssteen, University of North West). There are now nine official Bantu languages in South Africa, and so there's plenty of work still to do.

7. Conclusions

In conclusion, finite state techniques have been used to build morphological analyzer/generators not only for the obviously commercial languages but for an ever-increasing number of minority and lesser-studied languages. Morphological analysis is often the ideal first step when starting natural language processing, allowing you to use, test and even expand existing lexical resources. It's often a perfect project for a master's thesis.

Work on morphological analysis and generation leads directly to other lexical applications such as tokenization, baseform reduction, indexing, spelling checking and spelling correction. Morphological analyzers can also serve as reusable enabling components in larger systems that perform disambiguation, syntactic parsing, speech generation, speech recognition, etc.

Practical finite-state software is now widely available, with good documentation. In most cases, the software is leniently licensed and easily acquired for non-commercial use, with commercial licensing also possible.

It has been my privilege and pleasure to teach finite-state programming techniques to a number of linguists and computer scientists, and often to see them go home and produce sophisticated systems for exotic, lesser-studied natural languages that we have neither the time nor the expertise to handle at Xerox. Morphology projects are an exciting meeting point for descriptive linguists and computational linguists, often requiring collaboration, and giving each camp a better appreciation of what the other one does.

[23] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004L01

[24] http://cs.haifa.ac.il/~shuly/

[25] http://cs.haifa.ac.il/~shlomo/

[26] http://cs.haifa.ac.il/~yaelc/

[27] http://www.sics.se/humle/ile/kurser/Addis/amharic.shtml, http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/f/Fissaha:Sisay.html

[28] http://giellatekno.uit.no/

[29] http://www.ite.ie/morph.htm

8. References

Antworth, Evan L., 1990. *PC-KIMMO: a two-level processor for morphological analysis*. Number 16 in Occasional publications in academic computing. Dallas: Summer Institute of Linguistics.

Beesley, Kenneth R., 1992a. Klingon morphology, part 2: Verbs. *HolQeD*, 1(3):10–18.

Beesley, Kenneth R., 1992b. Klingon two-level morphology, part 1: Nouns. *HolQeD*, 1(2):16–24.

Beesley, Kenneth R., 2003. Finite-state morphological analysis and generation for Aymara. In *EACL 2003, 10th Conference of the European Chapter, Proceedings of the Workshop on Finite-State Methods in Natural Language Processing*. East Stroudsburg, PA: Association for Computational Linguistics.

Beesley, Kenneth R. and Lauri Karttunen, 2003. *Finite State Morphology*. Palo Alto, CA: CSLI Publications.

Johnson, C. Douglas, 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Kaplan, Ronald M. and Martin Kay, 1981. Phonological rules and finite-state transducers. In *Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting*. New York. Abstract.

Kaplan, Ronald M. and Martin Kay, 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Karttunen, Lauri, 1983. KIMMO: a general morphological processor. In Mary Dalrymple, Edit Doron, John Goggin, Beverley Goodman, and John McCarthy (eds.), *Texas Linguistic Forum, Vol. 22*. Austin, TX: Department of Linguistics, The University of Texas at Austin, pages 165–186.

Karttunen, Lauri, 1991. Finite-state constraints. In *Proceedings of the International Conference on Current Issues in Computational Linguistics*. Penang, Malaysia: Universiti Sains Malaysia.

Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen, 1992. Two-level morphology with composition. In *COLING'92*. Nantes, France.

Karttunen, Lauri, Kimmo Koskenniemi, and Ronald M. Kaplan, 1987. A compiler for two-level phonological rules. In Mary Dalrymple, Ronald Kaplan, Lauri Karttunen, Kimmo Koskenniemi, Sami Shaio, and Michael Wescoat (eds.), *Tools for Morphological Analysis*, volume 87-108 of *CSLI Reports*. Palo Alto, CA: Center for the Study of Language and Information, Stanford University, pages 1–61.

Kataja, Laura and Kimmo Koskenniemi, 1988. Finite-state description of Semitic morphology: A case study of Ancient Akkadian. In *COLING'88*.

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

Koskenniemi, Kimmo, 1984. A general computational model for word-form recognition and production. In *COLING'84*.

Koskenniemi, Kimmo, 1986. Compilation of automata from morphological two-level rules. In Fred Karlsson (ed.), *Papers from the Fifth Scandinavian Conference on Computational Linguistics*.

Lonsdale, Deryle, 2003. Two-level engines for Salish morphology. In *EACL 2003, 10th Conference of the European Chapter, Proceedings of the Workshop on Finite-State Methods in Natural Language Processing*. East Stroudsburg, PA: Association for Computational Linguistics.

Okrand, Marc, 1985. *The Klingon Dictionary:English/Klingon, Klingon/English*. New York: Pocket Books.

Roche, Emmanuel and Yves Schabes, 1997. Introduction. In *Finite-State Language Processing*, chapter 1. Cambridge, Massachusetts: MIT Press, pages 1–65.

Sproat, Richard, 1992. *Morphology and Computation*. Cambridge, MA: MIT Press.

Street, John C., 1963. *Khalkha Structure*. The Hague, Netherlands: Mouton.

Laying Lexical Foundations for NLP: the Case of Basque at the *Ixa* Research Group

Xabier Artola-Zubillaga

Ixa NLP Research Group
Fac. of Computer Science – Univ. of The Basque Country
649 p.k., 20080 Donostia
jiparzux@si.ehu.es

Abstract

The purpose of this paper is to present the strategy and methodology followed at the *Ixa* NLP Group of the University of The Basque Country in laying the lexical foundations for language processing. Monolingual and bilingual dictionaries, text corpora, and linguists' knowledge have been the main information sources from which lexical knowledge currently present in our NLP system has been acquired. The main lexical resource we use in research and applications is a lexical database, EDBL, that currently contains more than 80,000 entries richly coded with the lexical information needed in language processing tasks. A Basque wordnet has also been built (it has currently more than 50,000 word senses), although it is not yet fully integrated into the processing chain as EDBL is. Monolingual dictionaries have been exploited in order to obtain knowledge that is currently being integrated into a lexical knowledge base (EEBL). This knowledge base is being connected to the lexical database and to the wordnet. Feedback obtained from users of the first language technology practical application produced by the research group, i.e. a spelling checker, has also been an important source of lexical knowledge that has permitted to improve, correct and update the lexical database. In the paper, doctorate research work on the lexicon finished or in progress at the group is outlined as well, as long as a brief description of the end-user applications produced so far.

1 Introduction

Basque is a language spoken on both sides of the west-end border between France and Spain by approx. 700,000 people (25% of the population). It is co-official in some regions of the country, and moderately used in administration instances. Its use in education, from the mother school up to the university, is growing since the early eighties. There is one TV channel and a newspaper is published daily in Basque. The standardization of written language is in progress since 1968. More information on the web can be obtained at http://www.euskadi.net/euskara.

The purpose of this paper is to present the strategy and methodology followed at the *Ixa* NLP Group of the University of The Basque Country in laying the lexical foundations for natural language processing tasks.

In section 2, the *Ixa* NLP Research Group is introduced. Section 3 is devoted to describe EDBL, the main lexical database. Next two sections illustrate the construction of *Euskal Wordnet*, a wordnet of Basque, and EEBL, a lexical knowledge base which links the database, the wordnet, and knowledge derived from a monolingual dictionary. Next, in section 6, research work carried out at the group on the field of the lexicon is outlined. Finally, and before the conclusions, some end-user products and applications are briefly presented in section 7.

2 The *Ixa* NLP Research Group at the University of The Basque Country

The *Ixa* Research Group on NLP (http://ixa.si.ehu.es) belongs to the University of The Basque Country and its research has been conducted from the beginning (1986/87) on the fields of computational linguistics and language engineering.

The main application language has been and currently is Basque, but research and applications involving English, Spanish or French have been carried out as well. The strategy of language technology development at the group has been from the beginning a bottom-up strategy (Agirre *et al.*, 2001a, Díaz de Ilarraza *et al.*, 2003), that is, our goal has been first to lay the processing infrastructure –basic resources and tools–, in order to then be ready to produce end-user applications. Even our first product, a spelling checker, was conceived upon a general-purpose morphological analyzer (Alegria *et al.*, 1996).

The group is interdisciplinary (computer scientists and linguists) and it is formed nowadays by around 40 people, between lecturers, senior researchers and grant-aided students.

Ixa maintains scientific relationships with universities in different countries, and funding comes mainly from the university, local and Basque Governments, Spanish Government and European institutions.

3 EDBL: building the main lexical database from scratch

EDBL (Aldezabal *et al.*, 2001) is the name of our main lexical database, which is used as a lexical support for the automatic treatment of the language.

EDBL is a large store of lexical information that currently contains more than 80,000 entries. It has been conceived as a multi-purpose lexical basis, i.e. a goal-independent resource for the processing of the language.

The need of such a lexical database arose when the design and implementation of the morphological analyzer was faced. It was evident that a store for words and their attributes was necessary, and so, we took a dictionary and picked up all the entries with their part-of-speech (POS) information: this was the seed of EDBL. In these years, the design of the database has been significantly modified and updated in two occasions, to arrive to the current conceptual schema described in section 3.3.

3.1 The lexical database within the stream of language processing tools

EDBL is fully integrated in the chain of language processing resources and tools (see fig. 1), and the information contained in it is exported when required to be used as input by the language analysis tools.

A customizable exportation procedure allows us to select and to extract the information required by the different lexicons or tools in the desired format (XML, plain text, etc.). The lexicons obtained in this way are subsequently used in tools such as a morphological analyzer, a spelling checker (Aduriz *et al.*, 1997), a tagger/lemmatizer (Aduriz *et al.*, 1996a), and so on.

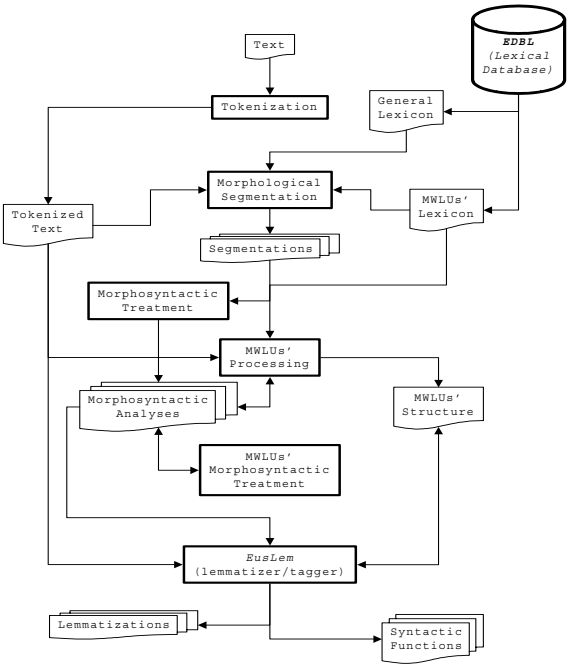


Figure 1: EDBL within the stream of language processing tools

3.2 Sources of knowledge used to populate EDBL

Different sources are used to populate the database: linguists and lexicographers' knowledge, monolingual and bilingual dictionaries, standard word lists regularly published by the Basque Language Academy (*Hiztegi Batua*: Euskaltzaindia, 2000), and the feedback given by the spelling checker (application and users) and other NLP tools such as the morphological analyzer or the lemmatizer.

When gaps in the database are detected, the lexicographer in charge of EDBL decides whether the entries are to be added or not, and fills the values for the required attributes. An especially conceived importation application facilitates this task to the lexicographer, allowing him or her to specify the input format, and making some deductions based on the POS of the entry, for example.

Apart from *Hiztegi Batua*, other dictionaries that have been used for this purpose are a small monolingual dictionary (Elhuyar, 1998), a Basque-Spanish/Spanish-Basque dictionary (Morris, 1998), a synonym dictionary (UZEI, 1999), and *Euskal Hiztegia* (Sarasola, 1996), a bigger monolingual explanatory dictionary.

3.3 Conceptual schema of the database

In this section, the Extended Entity-Relationship (EER) data model is used to describe the conceptual schema of the database (see fig. 2).

The main entity in EDBL is `EDBL_Units`, the key of which is composed of a headword and a homograph identifier, as in any conventional dictionary. Every lexical unit in EDBL belongs to this data class. The units in it can be viewed from three different standpoints, giving us three total specializations (all units in EDBL belong to the three specializations). This classifies every unit in EDBL into (1) standard or non-standard, (2) dictionary entry or other, and (3) one-word or multiword lexical unit.

Let us now have a glance at the three main specializations in the following subsections.

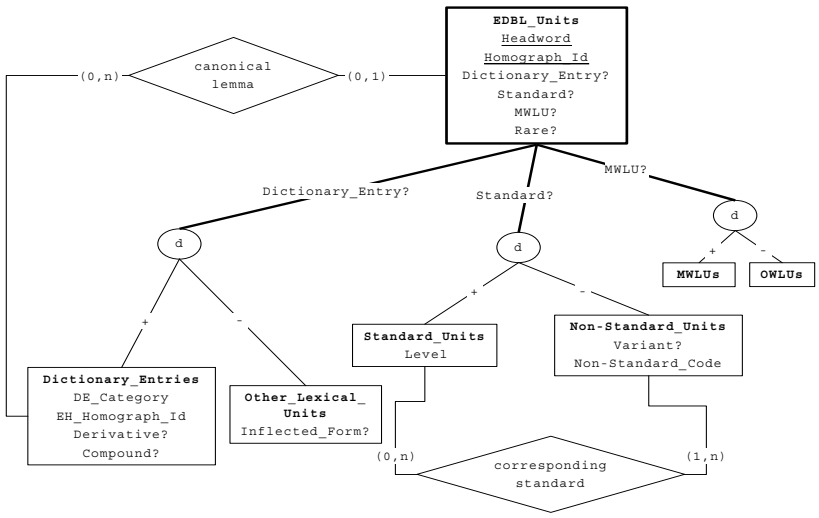


Figure 2: `EDBL_Units` and the three main specializations

3.3.1 Standard and non-standard lexical units

Basque is a language still in course of standardization; so, processes such as spell checking, non-standard language analysis, etc. require information about non-standard entries and their standard counterparts that must be stored in the lexical database, because, in fact, a relatively large number of non-standard forms may still be found in written language.

This specialization divides all the lexical units in EDBL into standard and non-standard. The entries belonging to the `Non-Standard_Units` class can be either variant (mainly dialectal) forms, or simply non-accepted entries.

The relationship between standard and non-standard units allows us to relate the correct forms to the ones considered incorrect. Each non-standard unit must be related at least to one standard unit.

3.3.2 Dictionary entries and other lexical units

Another main specialization in EDBL is the one that separates `Dictionary_Entries` from `Other_Lexical_Units`.

In the class of dictionary entries, we include any lexical entry that could be found in an ordinary dictionary, and they are further subdivided into nouns, verbs, adjectives, etc. according to their POS. Another specialization divides them into referential entries (symbols, acronyms, and abbreviations), compounds and derivatives.

On the other hand, `Other_Lexical_Units` is totally specialized into two disjoint subclasses: inflected forms and non-independent morphemes. Inflected forms are split up into verbal forms (auxiliary and synthetic verbs) and others (mostly irregularly inflected forms). Non-independent morphemes are affixes in general, which require to be attached to a lemma for their use inside a word form, and they are subdivided into different categories (graduators, declension morphemes, etc.).

Each class is characterized by different attributes. Nowadays these attributes are mainly of a morphosyntactic nature, although semantic features are already included in some cases.

3.3.3 One-word and multiword lexical units

The third total specialization of the main class classifies all the units in EDBL into One-Word Lexical Units (OWLUs) and Multiword Lexical Units (MWLUs). We consider an entry as OWLU if it has not any blanks in its spelling (hyphened forms and affixes included). Otherwise, it is taken as MWLU.

Every OWLU in EDBL is characterized by its morphotactics, i.e. the description of how it may be linked to other morphemes in order to constitute a word form. Being an agglutinative language, Basque presents a relatively high power to generate inflected word forms. Any entry independently takes each of the necessary elements (the affixes corresponding to the determiner, number and declension features) for the different functions (syntactic case included). This information is encoded in the database following the Koskenniemi's (1983) two-level formalism. So, our lexical system consists currently of 80,625 OWLUs, grouped into 201 two-level sublexicons and 159 continuation classes, and a set of 24 morpho-phonological rules that describe the changes occurring between the lexical and the surface level.

On the other hand, the description of a MWLU within the lexical database includes two aspects: (1) its composition, i.e. which its components are, whether they can be inflected or not, and according to which OWLU they inflect; and (2), what we call the surface realization, that is, the order in which the components may occur in the text, the components' mandatory or optional contiguousness, and the inflection restrictions applicable to each one of the components.

In that what concerns the surface realization, it is to be said that components of MWLUs can appear in the text one after another or dispersed; the order of components is not fixed, as some MWLUs must be composed in a restricted order while others may not: a MWLU's component may appear in different positions in the text; and, finally, the components may either be inflected (accepting any of the allowed inflection morphemes or in a restricted way) or occur always in an invariable form. Moreover, some MWLUs are "sure" and some are ambiguous, since it cannot be certainly assured that the same sequence of words in a text corresponds undoubtedly to a multiword entry in any context. According to these features, we use a formal description where different realization patterns may be defined for each MWLU.

3.4 Linguistic contents

We will give now some figures on the linguistic contents actually stored in EDBL.

According to the classification into the three main specializations, EDBL contains: 60,940 dictionary entries and 20,939 other lexical units (20,591 inflected forms and 348 non-independent morphemes); 78,417 standard forms and 3,462 non-standard; 80,625 OWLUs and 1,254 MWLUs. Among dictionary entries there are 40,087 nouns, 9,720 adjectives, 6,533 verbs, and 3,448 adverbs, among others; non-independent morphemes include 192 declension morphemes 45 subordinating morphemes, and 37 lexical suffixes, among others.

3.5 Current status and future improvements

At the *Ixa* group, we have designed and implemented a plan to integrate the exploitation of the language processing chain, in such a way that a common data exchange XML encoding is used as an input and delivery format between the different tools. According to this format, the information in the database is exported and delivered from it as a collection of feature structures.

So, the conceptual schema of the relational database has been mapped into a hierarchy of typed feature structures (FS). The leaves of this hierarchy are 22 disjoint classes, and each one of them defines a different FS type. When data are exported from EDBL, every EDBL unit is delivered into one of the 22 terminal FS types, including inherited features and others coming from nodes outside the main hierarchy. Information exported from EDBL is currently used in every task requiring morphological and/or syntactic processing.

In order to take advantage of all the information stored, our database has to be accessible and manageable. Even more, the fact that the users are not mainly computer scientists but linguists, stresses the reasons why we need a user-friendly, readily accessible and flexible interface. For that purpose, we designed and developed a graphic interface that gives help to the user based on context and that is accessible from the web (http://ixa2.si.ehu.es/edbl/). This GUI provides two levels of access to the database: one that lets common users only consult the data, and the second one that offers full reading and writing access, especially to the linguists in charge of the database.

4 *Euskal WordNet*: using bilingual and native dictionaries to construct a Basque wordnet

In EDBL, although homograph entries are separated, no semantic distinction between senses is made. As the group grew and the processing needs increased, semantics infrastructure became a must. As a point of departure, we decided to build *Euskal WordNet* (Agirre *et al.*, 2002), a Basque wordnet based on the English Wordnet (Fellbaum, 1998). Considering Wordnet as a *de facto* standard for the lexical-semantic representation for English, new wordnets in some other languages have been built, especially in the framework of the EuroWordNet project (EuroWN, http://www.illc.uva.nl/EuroWordNet/).

Euskal WordNet follows the EuroWN framework and, basically, it has been produced using a semi-automatic method that links Basque words to the English Wordnet (hereafter Wordnet). This section describes the current state of the Basque wordnet and the methodology we have adopted to ensure its quality in terms of coverage, correctness, completeness, and adequacy.

In order to ensure proper linguistic quality and avoid excessive English bias, a double manual pass on the automatically produced Basque synsets is desirable: a first concept-to-concept pass to ensure correctness of the words linked to the synsets, and then a word-to-word pass to ensure the completeness of the word senses linked to the words. By this method, we expected to combine quick progress (as allowed by a development based on Wordnet) with quality (as provided by a development based on a native dictionary). We have completed the concept-to-concept review of the automatically produced links for the nominal concepts, and are currently performing the word-to-word review.

4.1 Automatic generation and concept-to-concept review

In order to help the linguists in their task, we automatically generated noun concepts from machine-readable versions of Basque-English bilingual dictionaries (Morris, 1998; Aulestia & White, 1990). All English/Basque entry pairs in the dictionaries were extracted, and then were combined with Wordnet synsets; the resulting combinations were then analyzed following the class methods (Atserias *et al.* 1997). The algorithm produces triples like word - synset - confidence ratio. The confidence ratio is assigned depending on the results of the hand evaluation. The pairs produced by class methods with a confidence rate lower than 62% were discarded.

All the results of the previous process were validated by hand. The linguists reviewed the synsets that had a Basque equivalent one by one, checking whether the words were correctly assigned and adding new words to the synonym set if needed. This process led to the preliminary *Euskal WordNet 0.1* release.

4.2 Quantitative and qualitative analysis of *Euskal WordNet 0.1*

Table 1 reviews the amount of synsets, entries, etc. of the Basque wordnet compared to Wordnet 1.5 and the EuroWN final release (Vossen *et al.*, 2001). The first two rows show the number of Base Concepts, which were manually set. For nouns in the *Euskal WordNet 0.1*, the *Nouns (auto)* row shows the figures as produced by the raw automatic algorithm, and the *Nouns (man)* row the figures after the manual concept-to-concept review. The number of entries was manually reduced down to 50%, and the number of senses down to 15%. This high number of spurious entries and senses was caused primarily by a high number of orthographic and dialectal variants that were introduced by the older bilingual dictionary, which does not follow the standard current rules.

		Synsets	*No. of senses*	*Senses/ synset*	*Entries*	*Senses/ entry*
Euskal WordNet	**Nominal BC**	228	-	-	-	-
	Verbal BC	792	-	-	-	-
Euskal WordNet 0.1	**Nouns (auto)**	27641	291011	10.52	46164	6.3
	Nouns (man)	23486	41107	1.75	22166	1.8
	Verbs (man)	3240	9294	2.86	3155	2.95
Wordnet 1.5	**Nouns**	60557	107484	1.77	87642	1.23
	Verbs	11363	25768	2.27	14727	1.75
Dutch WordNet	**Nouns**	34455	54428	1.58	45972	1.18
	Verbs	9040	14151	1.57	8826	1.60
Spanish WordNet	**Nouns**	18577	41292	2.22	23216	1.78
	Verbs	2602	6795	2.61	2278	2.98
Italian WordNet	**Nouns**	30169	34552	1.15	24903	1.39
	Verbs	8796	12473	1.42	6607	1.89

Table 1: Figures for *Euskal WordNet 0.1* compared to Wordnet 1.5 and the final EuroWN release

The senses per entry figures are higher than those from Wordnet 1.5 and most of the wordnets, but similar to the Spanish WordNet. The fact that the nouns and verbs included are in general more polysemous can explain this fact. We also performed an analysis of the distribution for the variants in each synset and the number of word senses per entry.
All in all, the amount of synsets and entries for the *Euskal WordNet 0.1* is comparable to those for the wordnets produced in EuroWN, but lower than the Wordnet 1.5 release. The coverage of nominal concepts is 38% of those in Wordnet 1.5.
Somehow, we were not satisfied by the quantitative analysis and the results of the concept-to-concept review. On the one hand, the quantitative analysis only shows the state of the coverage of concepts and entries, as long as they are compared to reference figures from Wordnet (concepts) and Basque reference dictionaries (entries). It is rather difficult to assess the coverage of the number of word senses and synonyms, as these can only be compared to Wordnet, but there are no reference figures for the Basque wordnet itself. We think that the coverage of word senses and synonyms can be more reliably estimated measuring by hand the completeness of the word senses of a sample of words and the variants for a sample of concepts.
On the other hand, the concept-to-concept review only enforces the correctness and completeness of the variants in the synset. As the focus of the first stage was on quickly producing a first version, correctness was more important than completeness, and we were not completely satisfied with the completeness of the variants.
These are the correctness, completeness and adequacy requirements that were not covered by the quantitative analysis:

a) Correctness and completeness of the word senses of a word.
b) Correctness and completeness of the variants of a concept.
c) Adequacy of the specificity level for variants in a concept, i.e. all variants of a concept are of the same specificity level.
d) Adequacy of the specificity level for word senses, i.e. granularity of word senses.

In order to assess points a and d, we performed a manual comparison and mapping of the word senses given by *Euskal WordNet 0.1* with those of a monolingual dictionary and a bilingual dictionary. This assessment is presented in the next subsection.
We have also manually checked the correctness and completeness of the variants for a concept (b), using a synonym dictionary for this purpose. The results were highly satisfactory, but we decided to explicitly include the use of the synonym dictionary in all subsequent reviews and updates of the wordnet.

4.2.1 Manual mapping of word senses from the Basque wordnet and native dictionaries

The sense partition of any dictionary reflects a suitable native sense partition, and needs not to be of the same granularity as of Wordnet. In principle, both sense partitions could even be incompatible, in the sense that they could involve many-to-many mappings.
We chose to use the *Euskal Hiztegia* (EH) dictionary (Sarasola, 1996), a general purpose monolingual dictionary that covers standard Basque and that contains about 33,000 entries. One drawback of this dictionary is that it mainly focuses on literature tradition, and it lacks many entries and word senses which are more recent. For this reason, we decided to include also a bilingual Basque-English dictionary (Morris, 1998). Moreover, if the linguist thought that some other word sense was missing he/she was allowed to include it.
All in all, both bilingual and monolingual dictionaries contribute equally to the new senses. An average of 1.9 new senses are added for each word, which makes an average of 0.24 new senses for each existing sense. This makes an idea of the completeness of the word senses for words. All word senses were found to be correct. These figures can be interpolated to estimate that the coverage of word senses for the entries currently in *Euskal WordNet* is around 80%.
Regarding the mapping between the word senses of *Euskal WordNet* and the monolingual dictionary, most of the times it was one-to-one or many-to-one. The granularity of the word senses in *Euskal WordNet* is much finer. We have not found many-to-many mappings.

4.2.2 Adequacy of the specificity level of variants in synsets

As already mentioned in the quantitative analysis, we found out that some words had an unusually high number of senses. Quick hand inspection showed that for some concepts the variants were of heterogeneous specificity, and we suspected that some words were placed in too many concepts. In fact, a program that searches for words that have two word senses, one hypernym of the other, found out that there are 4,500 such pairs out of 41,107 word senses. This is a very high figure compared to Wordnet, and indicates that we need to check those word senses.

4.3 Conclusions of the quantitative and qualitative analysis and current status

We have presented here a methodology that tries to integrate the best of development methods based on the translation of Wordnet and development methods based on native dictionaries. We first have developed a quick core wordnet comparable to the final EuroWN release using semi-automatic methods that includes a concept-to-concept manual review, and later performed an additional word-to-word review based on native lexical resources that guarantees the quality of the wordnet.
As a summary of the quality assessment for the nominal part of the Basque wordnet, we can say that it contains 38% of the concepts in Wordnet 1.5, 25% of the entries (although it accounts for all the noun entries in EH), and 80% of the senses for the entries already in *Euskal WordNet*.

4.4 Word-to-word review and future work

Most of the shortcomings detected in the previous section can be overcome following an additional review

of the current *Euskal WordNet 0.1*. In this review we want to ensure that the coverage of word senses is more complete, trying to include the estimated 20% of word senses that are missing. In this case, the review is to be done studying each word in turn and taking attention to the following issues:

- Coverage of senses: add main word senses of basic words.
- Correctness of word senses of a word: delete inadequate word senses when necessary.
- Completeness of word senses of a word: add main word senses.
- Adequacy of the specificity level of word senses of a word: check that sense granularity is balanced.

The need to build a core wordnet led us to define a subset of the nominal entries to be covered: on the one hand, the top 400 words from a frequency analysis; on the other hand, the entries in a basic bilingual Basque-Spanish dictionary (Elhuyar, 1998) which defines a core vocabulary of Basque (13,000 nouns). The word senses are provided by the monolingual (EH) and the bilingual dictionaries. The bilingual dictionary includes modern words and word senses which are not in EH.

We are currently extending the coverage of the noun entries and word senses to those in a basic vocabulary of Basque. In the future we plan to apply the methodology to verbs and adjectives, and to extend the coverage to a more comprehensive set of nouns.

The current version of *Euskal WordNet* has about 25,400 entries and 52,500 senses that have been manually revised.

5 From the lexical database to a general-purpose lexical knowledge base: EEBL

A way to furnish EDBL with semantic content is to link it to other lexical resources such as machine-readable monolingual dictionaries (so providing it with definitions and related words), multilingual dictionaries (equivalents in other languages), etc.

This section describes a lexical-semantic resource under construction: EEBL, the Basque Lexical Knowledge Base, that constitutes the core of a research work currently in progress (Agirre *et al.*, 2003).

EEBL is a large store of lexical-semantic information that has been conceived as a multi-purpose and goal-independent resource for language processing tasks. It will be composed of three interlinked databases: EDBL, *Euskal WordNet*, and a dictionary knowledge base extracted from EH (see 5.1).

So, our aim here is to configure a general lexical-semantic framework for Basque language processing, linking EDBL entries with senses (definitions and examples) and related entries in the monolingual dictionary (derivatives, antonyms, hypernyms, hyponims, meronyms, etc.), synsets in *Euskal WordNet*, etc. On the other hand, this gives us the possibility to enrich the information contained both in the wordnet and in the dictionary knowledge base with the information contained in EDBL.

To start with, we decided to connect EDBL with the dictionary knowledge base and the last one with the wordnet. EDBL and *Euskal WordNet* have been already presented in this paper. In order to build a dictionary knowledge base from EH, word definitions in the dictionary have been semi-automatically analyzed to find and extract lexical-semantic relations among senses (see the next subsection). The results of such an analysis have been stored in the Concept Classification component of the EH Dictionary Knowledge Base (see fig. 3). It is worth underlining that criteria followed in the creation of both databases are quite different, and so are the obtained relations. Therefore, the integration (total or partial) of these databases allows mutual enrichment.

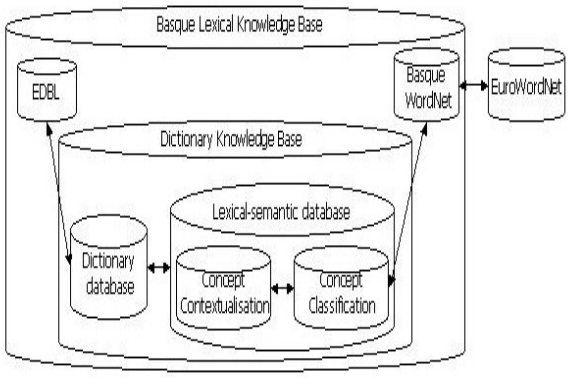


Figure 3: General architecture of the Basque Lexical Knowledge Base (EEBL)

The interrelation between EDBL and the EH Dictionary Knowledge Base allows us to manage lexical information of both grammatical and semantic nature, given that EDBL stores mainly grammatical information about words.

5.1 Exploiting a monolingual dictionary to build the EH Dictionary Knowledge Base.

The EH Dictionary Knowledge Base groups two different views of the dictionary data (see fig. 3). The Dictionary Database stores the dictionary itself in a conventional way whereas the Lexical-Semantic Database represents the lexical-semantic relations extracted from it in a semantic network-like fashion.

In a work currently in progress by Lersundi (Agirre *et al.*, 2000; Agirre & Lersundi, 2001), whose final goal is to enrich the lexical database with semantic information, EH (the monolingual dictionary) has been exploited to extract from it such kind of information. The work focuses on the extraction of the semantic relations that best characterize the headword, that is, those of synonymy, antonymy, hypernymy, and other relations marked by specific relators[1] and derivation.

All nominal, verbal and adjectival entries in EH have been parsed. Basque uses morphological inflection to mark case, and therefore semantic relations have to be inferred from suffixes rather than from prepositions. Our approach combines a morphological analyzer and

[1] We take as specific relators typical expression patterns used by the lexicographers when writing dictionary definitions. By means of these relators, some words in the definition text are linked to the headword in a special way, often determining the semantic relation that holds between them.

surface syntax parsing based on Constraint Grammar (Karlsson *et al.*, 1995), and has proven very successful for highly inflected languages such as Basque. Both the effort to write the rules and the actual processing time of the dictionary have been very low. At present we have extracted more than 40,000 relations, leaving only 9% of the definitions (mostly adjectives) without any extracted relation. The error rate is extremely low, as only 2.2% of the extracted relations are wrong.

The EH Dictionary Knowledge Base has been already supplied with the information extracted from EH. Namely, 33,102 dictionary units, 3,160 sub-entries (mainly multiword lexical units), and 45,873 senses with their corresponding relations are stored in the knowledge base.

In the future we plan to cover the semantic relations in the rest of the definition, that is, those relations involved in the part of the definition that is not the main defining pattern. For this we will use more powerful partial parsers (Aldezabal *et al.*, 1999). Besides, the coverage of derivational phenomena is also being extended, focusing specially in adjectival suffixes, in order to reduce the number of adjectives without any relation.

In order to include the extracted relations in EDBL (the lexical database), it is necessary to perform two disambiguation processes. On the one hand, there are some cases in which the surface relation extracted is ambiguous, that is, it could convey more than one deep semantic relation. On the other hand, the word senses of the words in the semantic relation have to be also determined. Anyway, some work aiming at the enrichment of EDBL based on the information extracted from EH has already been done. In particular, a method for semi-automatically assigning the animate feature to common nouns has been developed based mainly on the hyponym/hypernym relationships discovered in the dictionary (Díaz de Ilarraza *et al.*, 2002). The method obtains an accuracy of over 99% and a scope of 68,2% with regard to all the common nouns contained in a real corpus of over 1 million words, after the manual labelling of only 100 nouns. The results of this process have not yet been incorporated into EDBL.

5.2 Current status and future work

The level of integration between the lexical database and the EH Dictionary Knowledge Base can be summarized as follows: 80% of the total of entries in EH and 33% of the sub-entries have been satisfactorily linked to EDBL's entries. These links have been established automatically. In the case of derived entries, lemmatization has been used to establish links between roots, whenever it was not possible to link whole forms.

With respect to the lexical-semantic part of the knowledge base, the acquisition of relations from the dictionary is still in progress. Table 2 shows the number of relations that have been extracted from the dictionary and stored in the knowledge base so far. About 40,000 relations have been already stored. The difference between the number of extracted and stored relations is due, mainly, to the fact that some words occurring in definitions do not appear as entries. The other important reason is that some relations are duplicated because the morphological analyzer yields more than one single analysis for some words. In these cases, we only store one relation and avoid storing the same relation for different analyses.

	Extracted relations	**Stored relations**	
Synonyms	19,809	16,949	85.6%
Hypernyms	20,658	18,331	88.7%
Spec. relators	5,386	4,169	77.4%
Overall	45,853	39,449	86%

Table 2: State of the DKB

For the future we are planning to enhance the contents of the lexical-semantic framework. For this purpose we intend to:

- Deal with the relations extracted from a deeper analysis of the dictionary, including the derivational relationships.
- Repeat the same process with a bigger monolingual dictionary (Elhuyar, 2000).
- Include relations extracted from other sources, such as corpora, as it is aimed in the MEANING project at which the group is participating (Atserias *et al.*, 2004).
- Incorporate information on named entities and classify them.

6 Research completed and in progress

In this section, we would like to outline research work on the field of the lexicon, and particularly, to present doctorate research work carried out at the group on this field. Some of these works have been already completed while others are nearly finished or just in progress. In many of them, enrichment and improvement of the knowledge contained in the lexical database, especially in that what concerns its semantic component, is one of the main goals pursued. Different approaches and methodologies have been used for that.

Research work already finished includes:

- Artola's work (Artola, 1993; Agirre *et al.*, 1997) on a small French dictionary, where sense definitions were analyzed to semi-automatically extract lexical-semantic relationships. He proposes a general framework for the representation of dictionary knowledge, which is then used in a prototype of the so-called Intelligent Dictionary Help System (*Hiztsua*), a dictionary system aimed at human users.
- Following this work, Arregi (Arregi, 1995; Agirre *et al.*, 2001b) exports the idea to a multilingual system called *Anhitz*. In this system the representation model is extended and enriched to cope with a multilingual dictionary architecture. Arregi carried out as well a comprehensive and in-depth research on the use of dictionaries in translation tasks, so expanding the functionality of the system to a great detail.
- Agirre (Agirre & Rigau, 1996; Agirre, 1998) tackles the problem of word-sense disambiguation (WSD), and proposes a method for the resolution of lexical ambiguity that relies on the use of the Wordnet taxonomy and the notion of conceptual

distance among concepts, captured by a Conceptual Density formula developed for this purpose. This fully automatic method requires no hand coding of lexical entries, hand tagging of text nor any kind of training process.

- Arriola's work (Arriola *et al.*, 1999; Arriola, 2000) is motivated by two considerations: (1) the use of existing lexical resources to contribute to the design of more complete lexical entries, and (2) the acquisition of basic subcategorization information of verbs to support NLP tasks. The examples in verbal entries of the EH monolingual dictionary are analyzed in his work using for that a Constraint Grammar parser (Karlsson *et al.*, 1995), and basic subcategorization patterns are obtained.
- Aldezabal (Aldezabal *et al.*, 2002; Aldezabal, 2004) follows the previous work in the sense that she also looks for verb subcategorization information, which is an urgent need in our lexical system if we want to be able of performing deep syntactic parsing of free texts. In her thesis, Aldezabal makes an in-depth analysis of Levin's work (1993), and tries to adapt it to the case of Basque. As a result of this work, the occurrences of 100 verbs in a corpus have been thoroughly examined, and the different syntactic/semantic patterns applicable to each of them have been encoded in a database.

Research work currently in progress includes:

- Urizar's work (Aduriz *et al.*, 1996b), which is focused on the representation and processing of Multiword Lexical Units and multiword expressions in general. He proposes a representation schema for MWLUs that, due to its expressive power, can deal not only with fixed expressions but also with morphosyntactically flexible constructions. It allows to lemmatize word combinations as a unit and yet to parse the components individually if necessary. This work must be placed in a general framework of written Basque processing tools, which currently ranges from the tokenization and segmentation of single words up to the syntactic processing of general texts, and is closely related to the work by Ezeiza (Ezeiza, 2002), who developed a parser of multiword expressions.
- Martinez (Martínez *et al.*, 2002) explores the contribution of a broad set of syntactically motivated features to WSD. This set ranges from the presence of complements and adjuncts, and the detection of subcategorization frames, up to grammatical relations instantiated with specific words. The performance of the syntactic features is measured in isolation and in combination with a basic set of local and topical features, and using two different algorithms. Additionally, the role of syntactic features in a high-precision WSD system based on the precision-coverage trade-off is also investigated in his work.
- Atutxa's thesis (Aldezabal *et al.*, 2003) deals with lexical knowledge acquisition from raw corpora. The main goal is to automatically obtain verbal subcategorization information, using for that a shallow parser and statistical filters. The arguments are classified into 48 different kinds of case markers, which makes the system fine grained if compared to equivalent systems that have been developed for other languages. This work addresses the problem of distinguishing arguments from adjuncts, being this one of the most significant sources of noise in subcategorization frame acquisition.
- Finally, an architecture for a federation of highly heterogeneous lexical information sources is proposed in a PhD work nearly finished by Soroa (Artola & Soroa, 2001). The problem of querying very different sources of lexical information lexical and dictionary databases, heterogeneously structured electronic dictionaries, or even language processing programs such as lemmatizers or POS taggers, using for that a unique and common query language, is addressed in this work from the point of view of the information integration research field. The so-called *local-as-view* paradigm is used for describing each lexical source as a view over a general conceptual model. A general conceptual model for describing lexical knowledge has been designed, as well as the way to describe each source in terms of the classes and relationships of this general model. Both the conceptual model and the sources are described and implemented using a description logic language.

7 Products and applications

A first by-product of the research work accomplished on the field is *Xuxen*, a morphological analysis based general-purpose spelling checker/corrector (Aduriz *et al.*, 1997) widely used nowadays.

Moreover, two dictionaries have been also integrated as plugins into *Microsoft Word*: a Basque-Spanish bilingual dictionary (Elhuyar, 1998) and a synonym dictionary (UZEI, 1999); in both cases on-the-fly lemmatization is performed when consulting them, allowing users a very handy lookup.

The forthcoming publication of a quite sophisticated electronic version of *Euskal Hiztegia* (Arregi *et al.*, 2003), a monolingual dictionary already mentioned several times in this paper, which has been parsed from its original RTF format and encoded into XML following the TEI guidelines, completes the panorama of end-user applications co-published by the group. This electronic version of the dictionary allows the user to search into the definitions and examples as in a fully lemmatized corpus, by posing complex queries based on lemmas and/or inflected forms, and using logical operators to construct the queries.

8 Conclusions

A language that seeks to survive in the modern information society requires language technology products. "Minority" languages have to make a great effort to face this challenge. Lesser-used language communities need, in our opinion, a long-term and well-thought strategy if they want to be able to produce language technology applications; good foundations in

terms of resources and basic tools are a must to get this goal.

At the *Ixa* NLP Group, the development of language technology has been faced from the very beginning in a bottom-up fashion, that is, laying first the infrastructure (resources and tools) in order to later be able to produce end-user applications. If anything, it is principally this conception of the strategy we have designed and developed that we could "export" to other "minority" languages as ours.

Based on our 15-year experience in NLP research, we can conclude that the combination of (semi-)automatic procedures and manual work warrants a moderately fast but reliable setting when building the lexical foundations needed in NLP. Common dictionaries constitute an obvious resource for NLP: lists of words, homographs and senses, basic grammatical information (POS, subcategorization, etc.), and, if further worked out, lots of implicit knowledge about words and their interrelationships may be extracted from them.

We have shown that work done for "bigger" languages has been sometimes very useful for our research: the use of the English Wordnet along with bilingual dictionaries has facilitated our work when building the Basque wordnet. However, if NLP research is conducted only on the "main" languages, there will be nothing we can do about the survival of our "minor" languages. Investigation on the language itself and on the application of general techniques to the processing of the language are needed as well, and, moreover, they contribute to general research in the sense that they provide a different and enriching point of view of the problems undertaken.

In the paper just our work on laying the lexical infrastructure for NLP has been presented. We are currently working as well on other areas of NLP, ranging from morphology to semantics, and tackling problems related to machine translation, computer-aided language learning, information retrieval and extraction, etc.

Basque is a minority language but we think that a substantial amount of work has already been done in the field, and that sound foundations have been established. As it has been said above, we firmly believe that the establishment of such an infrastructure is fundamental for language technologies to be developed, and our group is entirely devoted to this research since its inception in the late eighties. To finish, just to say that, apart from us, several other groups are also working on Basque automatic processing; we think that the cooperation between the different groups and sharing of the results should undoubtedly improve the development of the whole field in our country.

9 References

Publications by the Ixa Group

Aduriz I., Aldezabal I., Alegria I., Urizar R. (1996a). EUSLEM: A lemmatiser/tagger for Basque. EURALEX'96, Gotteborg (Sweden).

Aduriz I., Aldezabal I., Artola X., Ezeiza N., Urizar R. (1996b). Multiword Lexical Units in EUSLEM, a lemmatiser/tagger for Basque. Proc. of the 4th Conf. on Computational Lexicography and Text Research, COMPLEX'96, Linguistics Institute, Hungarian Academy of Science. Budapest (Hungary).

Aduriz I., Alegria I., Artola X., Ezeiza N., Sarasola K., Urkia M. (1997). A Spelling Corrector for Basque Based on Morphology. Literary and Linguistic Computing 12/1, pp. 31-36. ALLC, Oxford (England).

Agirre, E., Rigau, G. (1996). Word Sense Disambiguation using Conceptual Density, Proceedings of COLING'96, pp. 16-22. Copenhagen (Denmark).

Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., Soroa A. (1997). Constructing an intelligent dictionary help system. Natural Language Engineering 2(3): 229-252. Cambridge Univ. Press. Cambridge (England).

Agirre, E. (1998). *Kontzeptuen arteko erlazio-izaeraren formalizazioa ontologiak erabiliaz: Dentsitate Kontzeptuala*. / *Formalization Of Concept-Relatedness Using Ontologies: Conceptual Density*. PhD thesis. Dept. of Computer Languages and Systems, University of The Basque Country.

Agirre E., Ansa O., Arregi X., Artola X., Díaz de Ilarraza A., Lersundi M., Martínez D., Sarasola K., Urizar R. (2000). Extraction of semantic relations from a Basque monolingual dictionary using Constraint Grammar. Proceedings of Euralex. Stuttgart (Germany).

Agirre E., Aldezabal I., Alegria I., Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Sarasola K., Soroa A. (2001a). Developing Language Technology for a Minority Language: Progress and Strategy. Elsnews 10.1, pp. 4-5. ELSNET, Utrecht (The Netherlands).

Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., Soroa A. (2001b). MLDS: A Translator-Oriented Multilingual Dictionary System. Natural Language Engineering, 5 (4), pp. 325-353. Cambridge Univ. Press. Cambridge (England).

Agirre, E., Lersundi, M. (2001). Extracción de relaciones léxico-semánticas a partir de palabras derivadas usando patrones de definición. In Proceedings of SEPLN 2001. Jaén (Spain).

Agirre E., Ansa O., Arregi X., Arriola J.M., Díaz de Ilarraza A., Pociello E., Uria L. (2002). Methodological issues in the building of the Basque WordNet: quantitative and qualitative analysis. Proceedings of the First International WordNet Conference. Mysore (India).

Agirre E., Ansa O., Arregi X., Artola X., Díaz de Ilarraza A., Lersundi M. (2003). A Conceptual Schema for a Basque Lexical-Semantic Framework. Proceedings of COMPLEX'03. Budapest (Hungary).

Aldezabal, I., Gojenola, K., Oronoz, M. (1999). Combining Chart-Parsing and Finite State Parsing, Proceedings of the European Summer School in Logic, Language and Information (ESSLLI) Student Session. Utrecht (The Netherlands).

Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernández G., Lersundi, M. (2001). EDBL: a General Lexical Basis for The Automatic Processing of Basque. IRCS Workshop on Linguistic Databases. Philadelphia (USA).

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K. (2002). Learning Argument/Adjunct

Distinction for Basque. ACL'2002 SigLex Workshop on Unsupervised Lexical Acquisition. Philadelphia (USA).

Aldezabal I., Aranzabe M., Atutxa A., Gojenola K., Sarasola K. (2003). A unification-based parser for Basque and its application to the automatic analysis of verbs. *In* B. Oyharçabal ed., *Inquiries into the lexicon-syntax relations in Basque*. Supplements of ASJU no. XLVI.

Aldezabal I. (2004). *Aditz-azpikategorizazioaren azterketa sintaxi partzialetik sintaxi osorako bidean. 100 aditzen azterketa, Levin-en (1993) lana oinarri hartuta eta metodo automatikoak baliatuz.* PhD thesis. Dept. of Basque Filology, University of The Basque Country.

Alegria I., Artola X., Sarasola K., Urkia M. (1996). Automatic Morphological Analysis of Basque. Literary and Linguistic Computing 11/4, pp. 193-203. ALLC, Oxford (England).

Arregi, X. (1995). *ANHITZ: Itzulpenean laguntzeko hiztegi-sistema eleanitza*. PhD thesis. Dept. of Computer Languages and Systems, University of The Basque Country.

Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., García E., Lascurain V., Sarasola K., Soroa A., Uria L. (2003). Semiautomatic construction of the electronic *Euskal Hiztegia* Basque Dictionary. *In* Traitement automatique des langues. Les dictionnaires électroniques (ed. Michael Zock and John Carroll), 44-2, pp. 107-124. ATALA. Paris (France).

Arriola J.M., Artola X., Maritxalar A., Soroa A. (1999). A Methodology for the Analysis of Verb Usage Examples in a Context of Lexical Knowledge Acquisition from Dictionary Entries. Proc. of EACL'99. Bergen (Norway).

Arriola J.M. (2000). *Euskal Hiztegiaren azterketa eta egituratzea ezagutza lexikalaren eskuratze automatikoari begira. Aditz-adibideen analisia Murriztapen Gramatika baliatuz, azpikategorizazioaren bidean*. PhD thesis. Dept. of Basque Filology, University of The Basque Country.

Artola, X. (1993). *HIZTSUA: Hiztegi-sistema urgazle adimendunaren sorkuntza eta eraikuntza. Hiztegi-ezagumenduaren errepresentazioa eta arrazonamenduaren ezarpena. / Conception et construction d'un système intelligent d'aide dictionnariale (SIAD). Acquisition et représentation des connaissances dictionnariales, établissement de mécanismes de déduction et spécification des fonctionnalités de base.* PhD thesis. Dept. of Computer Languages and Systems, University of The Basque Country.

Artola X., Soroa A. (2001). An Architecture for a Federation of Highly Heterogeneous Lexical Information Sources. IRCS Workshop on Linguistic Databases. Philadelphia (USA).

Díaz de Ilarraza A., Mayor A., Sarasola K. (2002). Semiautomatic labelling of semantic features. Proc. of COLING'2002. Taipei (Taiwan).

Díaz de Ilarraza A., Sarasola K., Gurrutxaga A., Hernaez I., Lopez de Gereñu N. (2003) HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities. Workshop on NLP of Minority Languages and Small Languages (TALN). Nantes (France).

Ezeiza, N. (2002) *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile morfosintaktiko sendo eta malgua*. PhD thesis. Dept. of Computer Languages and Systems, University of The Basque Country.

Martínez D., Agirre E., Márquez L. (2002). Syntactic features for high precision Word Sense Disambiguation. Proc. of the 19th International Conf. on Computational Linguistics (COLING 2002). Taipei (Taiwan).

Other references

Atserias J., Climent S., Farreras J., Rigau G., Rodríguez, H. (1997). Combining Multiple Methods for the Automatic Construction of Multilingual WordNets. In Proceedings of Conference on Recent Advances on NLP. (RANLP'97). Tzigov Chark (Bulgaria).

Atserias J., Villarejo L., Rigau G., Agirre E., Carroll J., Magnini B., Vossen P. (2004). The MEANING Multilingual Central Repository. Proc. of the 2nd Global WordNet Conference. Brno (Czech Republic).

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge (Massachusetts, USA). London (England).

Karlsson F., Voutilainen A., Heikkila J., Anttila A. eds. (1995). *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Berlin and New York: Mouton de Gruyter.

Koskenniemi K. (1983). *Two-level Morphology: A general Computational Model for Word-Form Recognition and Production*. PhD thesis, University of Helsinki (Finland).

Levin B. (1993). *English Verb Classes and Alternations. A Preliminary Investigation.* Chicago and London. The University of Chicago Press.

Vossen P., Bloksma L., Climent S., Marti M.A., Taule M., Gonzalo J., Chugur I., Verdejo F., Escudero G., Rigau G., Rodriguez H., Alonge A., Bertagna F., Marinelli R., Roventini A., Tarasi L., Peters W. (2001). Final Wordnets for Dutch, Spanish, Italian and English, EuroWordNet (LE2-4003) Deliverable D032/D033, University of Amsterdam (The Netherlands).

Dictionaries

Euskaltzaindia (2000). *Hiztegi Batua*. Euskera 5, Bilbo. http://www.euskaltzaindia.net/hiztegibatua/

Elhuyar (1998). *Elhuyar Hiztegi Txikia.* Elhuyar Kultur Elkartea, Usurbil.

Morris, M. (1998). *Morris Hiztegia*. Klaudio Harluxet Fundazioa, Donostia. http://www1.euskadi.net/morris/indice_e.htm

UZEI (1999). *Sinonimoen Hiztegia.* UZEI, Donostia. http://www.uzei.org/nagusia.cfm?hizkuntza=0&orria=online&atala=sinonimoak

Sarasola, I. (1996). *Euskal Hiztegia*. Kutxa Fundazioa, Donostia.

Aulestia, G., White, L. (1990). *English-Basque Dictionary*. University of Nevada Press, Reno (USA).

Elhuyar (2000). *Hiztegi Modernoa.* Elhuyar Kultur Elkartea, Usurbil.

First steps in corpus building for linguistics and technology

Dafydd Gibbon

Fakultät für Linguistik und Literaturwissenschaft
Universität Bielefeld, Germany

Abstract

A corpus is one of several possible sources of spoken, written or multimodal language data, there are many kinds of corpus, a corpus can be a rather complex entity, and a description of a corpus is multidimensional. In this introduction, relevant steps for corpus building are discussed, focussing particularly on corpus metadata, and on designing, creating, and processing corpora in the contexts of language documentation and the human language technologies. Particular attention is paide to requirements for corpus building for endangered languages. Examples will be taken from African languages to illustrate specific cases of corpus building and processing. Essential background is given in (Gibbon et al., 1997) and (Gibbon et al., 2000).

1. Introduction and overview

Text and speech corpora form the empirical grounding of linguistics, phonetics, the human language technologies, and many related disciplines. In this contribution I will start by asking some "frequently asked" questions about corpora, some fake and some genuine, with a couple of clarificatory footnotes. Then I will outline five steps in bulding corpora: finding standards, defining corpora, characterising corpora, classifying corpora (metadata), and finally the three phases of in corpus creation. The definitional step introduces three key criteria, and outlines the position of corpora among other text and speech resources. Finally I discuss two cases of corpus building: corpora in the documentation of an endangered language, and the newly raised issue of securing the interpretability of resources (including corpora).

2. Some initial questions

What is a corpus? Informally, a corpus is a homogeneous collection of written texts or recorded speech,[1] though this definition is not sufficient. If a corpus is defined as 'a homogeneous collection of texts', then does this include pile of old newspapers and magazines that you were meaning to read through before finally throwing away? Maybe, but only if certain other conditions are fulfilled. These will be discussed below. In this contribution I will concentrate on speech corpora.

Who needs a corpus? First and foremost: every linguist, natural language processing specialist and every speech technologist needs a corpus. Why? Because a corpus provides the empirical data for language and speech processing in all scientific, technological, office and hobby contexts.[2]

Why do I need a corpus? If a text corpus is meant, for example, your application could be a dictionary, or a language-learning textbook, or a grammar handbook, either in print or in a hypertext medium. If a speech corpus is meant, the application could be a machine-usable grammar or dictionary, a speech synthesiser, a speech recogniser, or a complex piece of software like a dictation enabled word processor with spell checker. Or it could be the construction of a heritage archive for an endangered unwritten language spoken in the tropics.

What kind of corpus do I need? First your needs must be defined. For written texts, perhaps a major need is in the area of lexicography, but specific needs depend on the kind of lexicon being developed, for example a print lexicon (of which there are many kinds), or a machine-readable lexicon for use in a word processor or automatic dictation environment. For speech, unit-selection speech synthesis and automatic speech recognition are typical applications.

Where do I get a corpus? In simple terms, you can make your own, hire an expert to make one for you, or buy one from an international language resource agency such as LDC or ELRA/ELDA.

How long does it take to make a corpus? With questions like this my grandfather used to answer with another question: 'How long is a piece of string?' It depends on your requirements. Time, material and human resources in the logistics of corpus building can vary greatly. For qualitative analysis, a corpus can be quite small, for example somewhere between a few minutes and an hour or so of speech. For extensive quantitative analysis with sophisticated statistical, information retrieval and text-mining technologies it could amount to many millions of words. An endangered language corpus can involve a lengthy stay of months or years in the region, if speakers of the language are not available outside their local area, or if an anthropologically interesting interactive audiovisual corpus is needed.

How much will I have to pay for a corpus? Again, it

[1] It is quite common in the human language technologies to use the terminology 'language' for written language, i.e. texts, and 'speech' for spoken language. This can be very misleading in an interdisciplinary context, so I will use the terminology 'text' for written language and 'speech' for spoken language. I will use 'multimodal communication' to refer to combinations of human input and output modalities (e.g. manual-visual for writing, oral-auditory for speech, etc.) in a single communication situation, and 'multimedia' to refer to combinations of natural and electronic media in these situations. See (Gibbon et al., 2000).

[2] The term 'corpus', from Latin 'corpus', is etymologically related to words like 'corps', 'corpse', 'corporation', and means *body*. The plural is 'corpora'; I have sometimes heard 'corpuses', which to a linguist sounds like the over-regular speech of a 3-year-old. It is well within the capabilities of professional processing experts and spell checkers to remember irregular plurals and their origins...

depends, this time on how much you want to pay. For speech, plain orthographic transcription or orthographic annotation takes around 50 times real time, that is, for an hour of speech you will need to pay for 50 hours of transcriber time. Transcription and annotation at the phonemic level will take more time, and detailed phonetic transcription a great deal more time; the same applies to multimodal annotation and transcription. Since continuous transcription is a very strenuous and demanding task, this will amount to about two weeks work per hour of recorded speech in normal working conditions. For writing, much will depend on the copyright situation.

Can I sell my corpus? If someone wants to buy, it, sure. The best strategy would be to consult one of the national or international resource dissemination agencies.

How can people access my corpus? There is a growing tendency to treat corpora as *open resources*. This does not necessarily mean that the corpus itself costs nothing, though in many cases this would be the ideal situation. But, realistically, corpora are expensive to make and consequently their creators understandably hesitate to make the corpora themselves freely available if they themselves have funding problems. The minimal case is to provide *Open Metadata*, that is, freely available information about the corpus, generally via an internet portal. The *Open Archive Initiative* (OAI) involves large, generally public institutions such as libraries and other archives; the *Open Language Archive Community* (OLAC) provides a metadata portal for resources for the study of language, whether in linguistics, phonetics, or in the human language technologies. These initiatives are easy to find by means of a straightforward internet search, so it is not necessary to give specific URLs here.

3. Step 1: Finding standards for corpora

It is perhaps not obvious that taking standards into account should be the first step in developing a corpus. Standards are sometimes regarded as an inhibiting nuisance (and sometimes they are), but in general, apart from their prescriptive nature in commercial quality control and market dominance contexts, they embody a wide range of experience and expert discussion which can often be taken as 'best practice' in a given area. As in any area of creative activity, standards form the benchmark against which further developments are measured; in many areas of research and development they are not immediately relevant, but when notions like interoperability and reusability are focussed, their relevance re-emerges.

There are many layers of standardisation, from local laboratory conventions through *de facto* standards (including industrial items like PCs, operating systems and office software) to institutionally defined and agreed national and international standards.

In the area of human language technologies, standardisation has consistently been a hot topic for around two decades, starting with the SAM project, coordinated by Adrian Fourcin in the 1980s, and continuing with the two phases of the EAGLES project and the recent ISLE project, coordinated by Antonio Zampolli in the 1990s. There are several ISO committees currently concerned with formulating agreements on standards for human language technology applications, terminology, annotation and transcription, and many other issues. The functionality of standardisation is to ensure interoperability in the case of tools, and re-usability in the case of language resources; the term *reusable resources* was coined by by Antonio Zampolli in the early 1990s, and formed the basis for a range of European funded projects in that decade.

Relatively recently, issues of corpus standards and resources as developed in the human language technologies (Gibbon et al., 1997; Gibbon et al., 2000; Bird and Liberman, 2001) have been extended to fieldwork corpora in linguistics, ethnography, and related sciences.

Further issues such as the role of metadata in resource archiving, reusability, the use of XML and Unicode based standardisation-friendly technologies, and the standards for open language archives have been emerging in recent years and will require addition to and revision of existing standards as these approaches mature.

4. Step 2: Defining corpora

Based on existing standards, and within the context of project-specific needs, corpus requirements can be defined, starting with the question: What is a corpus?

Technically speaking, a text or speech corpus is a collection of tokens of writing or speech which is either custom-made or compiled from existing texts or speech, intended for use in the disciplines of corpus linguistics, computational linguistics, natural language processing, or speech technology, and is accompanied by a characterisation which makes the corpus interpretable to and manageable by the user.

According to this definition, a corpus has three defining aspects:

1. A corpus is a *collection* of tokens.

2. A corpus has a *function* in language and speech processing.

3. A corpus requires interpretation via a *characterisation* (which includes *metadata*).

Tokens of writing (sometimes called 'inscriptions') may be inscribed on an analogue visual and spatial artefact such as paper, as handwriting or print, or in electronically coded digital form. Tokens of speech may be on an analogue medium such as a reel or cassette tape or a shellac or vinyl disk (depending on age), or in a sampled digital format.

Texts are often seen as spatial functions, while speech is often seen as a temporal function. Technically, however, these perspectives are functions of the media used: the production of a text is a temporal function of human behaviour, whether handwriting or keyboards are involved, and a recorded speech signal can be seen as a spatial function, just like recorded writing.

Not all collections of texts and speech are conventionally regarded as corpora. For instance a library, or the world-wide web, are not corpora in this sense, though they can be - and are being - increasingly utilised as such.

There are as many kinds of corpus as there are uses of text and speech and ways of describing and processing them.

A well-known list of distinctions between text and speech or multimodal corpora, originally due to Hans Tillmann, is the following (Gibbon et al., 1997), p. 81:

1. durability of text as opposed to the volatility of speech,
2. different time taken to produce text and speech,
3. different roles played by errors in written and spoken language,
4. differences in written and spoken words,
5. different data structures of ASCII (or otherwise encoded) strings and sampled speech signals,
6. great difference in storage size between text and speech or multimodal data collections,
7. different legal and ethical status of written text and spoken utterances,
8. fundamental distinction (and relation) between symbolically specified categories and physically measured time functions.

These distinctions highlight important criteria for corpus definition and the procedure of corpus building which need to be worked out carefully for each invididual corpus.

5. Step 3: Corpus characterisation

A collection of tokens has great potential as a corpus, but needs to be *interpretable*. The general term for techniques of securing interpretability in the case of corpus resources is *corpus characterisation*, which involves both linguistic and physical characterisation (Gibbon et al., 1997).

Linguistic characterisation involves the following components:

1. corpus units, i.e. distribution of token units in the corpus, such as phonemes in natural or balanced corpora;
2. corpus lexicon, i.e. a word list containing word types, token frequencies, type-token ratio and type-token ratio saturation as a corpus increases in size, and lists of tokens paired with their contexts of occurrence in the corpus (otherwise known as *concordances* (Gibbon and Trippel, 2002);
3. corpus grammar, i.e. a linguistic sketch grammar constructed with qualitative linguistic methods, or a stochastic language model (as used in speech recognition systems) constructed purely with statistical distributional analysis methods;
4. corpus situation, i.e. characterisation of the speaker, the utterance situation, the dialectal, social or functional language variety.

Increasingly, machine learning methods of automatic corpus analysis and induction of lexical and grammatical generalisations are being used to support the characterisation of a corpus; see contributions to (van Eynde and Gibbon, 2000). It may be noted in passing that the expectation of fully standardising the entire metadata specification tends to reveal singularly little awareness of the potential of machine learning and text mining procedures for handling generalisation tasks of this kind. It may be predicted that such procedures will be applied in future not only to extensive resource data sets but also to increasingly extensive sets of metadata.

Physical characterisation involves other requirements:

1. hardware and software format specification;
2. processing software functionality specification;
3. signal specification, including properties of microphone and other components in the signal chain, and analog or digital signal storage formats;
4. text format specification, through a hierarchy of text objects from characters to hypertext documents;

A useful taxonomy for corpora within the overall context of text and wpeech resources, one of many possible taxonomies, is the following:

- Speech resource objects:
  - Signal objects: audio, visual or other (e.g. laryngograph, airflow) recordings.
  - Signal processing objects: filter operations, fourier transformations, etc., on signal recordings.
- Text resource objects
  - Linguistically relevant written language text genres:
    - Grammars.
    - Dictionaries (concordances, alphabetic dictionaries, stem dictionaries, thesauri).
    - Texts (of all kinds, from technical to everyday to literary texts).
  - Linguistically relevant spoken language text genres (these are essentially *metalinguistic* and incorporate - usually implicitly - theoretical linguistic and phonetic assumptions):
    - Spoken language transcriptions (orthographic or in phonetic alphabets and multimodal transcription systems.
    - Spoken language annotations (i.e. transcriptions plus timestamps, in various degrees of granularity and on various parallel tiers).

Table 1: Example of OLAC XML code.

```
<oai:record>
<oai:header>
  <oai:identifier>oai:langdoc.uni-bielefeld.de:UBI-EGA-004</oai:identifier>
  <oai:datestamp>2003-10-31</oai:datestamp>
  </oai:header>
<oai:metadata>
<olac:olac xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
    xmlns="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xsi:schemaLocation="http://www.language-archives.org/OLAC/1.0/
        http://www.language-archives.org/OLAC/1.0/olac.xsd
        http://purl.org/dc/elements/1.1/
        http://www.language-archives.org/OLAC/1.0/dc.xsd
        http://purl.org/dc/terms/
        http://www.language-archives.org/OLAC/1.0/dcterms.xsd">
<title>Audio files for West African Language Data Sheets; Standard
    WALDS questionnaire for Ega </title>
  <creator>Baze, Lucien</creator>
  <subject>Audio files for WALDS questionnaire for Ega</subject>
  <language xsi:type="olac:language" olac:code="x-sil-EGA"/>
  <description>Audio files for WALDS questionnaire for Ega.</description>
  <identifier>
    http://www.spectrum.uni-bielefeld.de/langdoc/EGA/OLAC/Resources/Audio/MP3/
  </identifier>
  <publisher>unpublished</publisher>
  <contributor xsi:type="olac:role" olac:code="editor">Baze, Lucien</contributor>
  <contributor xsi:type="olac:role" olac:code="editor">Gibbon, Dafydd</contributor>
  <type xsi:type="olac:linguistic-type" olac:code="lexicon"/>
  <format>mp3</format>
  <format>wave file (available on request)</format>
  <source>Recordings of West African Language Data Sheets questionnaire
    elicitation sessions</source>
  <language xsi:type="olac:language" olac:code="x-sil-FRE"/>
  <language xsi:type="olac:language" olac:code="x-sil-ENG"/>
  <language xsi:type="olac:language" olac:code="x-sil-EGA"/>
  <relation>oai:langdoc.uni-bielefeld.de:UBI-EGA-002</relation>
  <coverage>Cote d'Ivoire</coverage>
</olac:olac>
</oai:metadata>
</oai:record>
```

6. Step 4: Classifying corpora: metadata

Linguistic and physical characterisation feed into the definition of *metadata*, i.e. description of the immediate properties of the corpus which are minimally necessary for identifying the corpus and its basic properties. Metadata for data resources in general correspond, roughly, to catalogue information for publications, e.g. for books in libraries. The main function of metadata is to enable efficient and, within practical limits, comprehensive search of language archives.

Metadata are conventionally formulated for any type of language resource, in addition to actual publications, i.e.

- spoken language corpora,
- written language corpora,
- lexica,
- grammars.

Several approaches to standardising metadata for different tasks have been developed during the past decade. All of these can easily be located on the internet. The most well-known is the Dublin Core (DC) metadata standard, which defines metadata categories such as the following: *Title*, *Publisher*, *Description*, *Language*, *Source*, *Contributor*, *Author*, *Subject*, *Subject*, *Date*, *Type*, *Format*, *Identifier*, *Relation (to other documents)*, *Coverage*, *Rights*.

The DC metadata set was basically designed for library-type information about texts. The Open Language Archive Community (OLAC) uses the Dublin Core set, enhanced with specific linguistically relevant elements which are not covered by the DC set but which are useful for linguistic purposes. An example of the use of OLAC metadata elements in a metadata record is shown in Table 1; the metadata refers to data for the endangered language Ega, an endangered language spoken in southern central Ivory Coast.

Other sets such as the IMDI (ISLE Metadata Initiative)

set have been or are being developed for linguistics.

Table 2: UBIcorpus metadata specifications.

Attribute	Type
RecordID:	string
LANGname(s):	popup: Agni,Agni; Ega
SILcode:	popup: ANY; DIE
Affiliation:	string
Lect:	string
Country:	popup: Côte d'Ivoire
ISO:	popup: CI
Continent:	popup: Africa; AmericaCentral; AmericaNorth; AmericaSouth; Asia; Australasia; Europe
LangNote:	longstring
SESSION:	popup: FieldIndoor; FieldOutdoor; Interview; Laboratory
SessionDate:	pick
SessionTime:	pick
SessionLocale:	string
Domain:	popup: Phonetics; Phonology; Morphology; Lexicon; Syntax; Text; Discourse; Gesture; Music; Situation
Genre:	Artefacts; Ceremony; Dialogue; Experiment-Perception; ExperimentProduction; History; Interview; Joke/riddle; Narrative; Questionnaire; Task
Part/Sex/Age:	string
Interviewers:	string
Recordist:	string
Media:	popup: Airflow; AnalogAudio; AnalogAV; AnalogStill; AnalogVideo; DigitalVideo; DigitalAudio; DigitalAV; DigitalStill; DigitalVideo; Laryngograph; Memory; Paper
Equipment:	longstring
SessionNote:	longstring

In my own linguistic fieldwork I have found that these metadata sets are inadequate for the purpose. For this reason I designed a metadata set for audio/video recordings, photos, paper notes and artefact cataloguing, the UBIcorpus (University of Bielefeld) metadata set. For convenience of use in fieldwork, a metadata editor was developed for the Palm handheld computer, using the HanDBase relational database management system. The metadata editor provides a fast and inconspicuous input method for structured metadata for recordings and other field documentation. The metadata set used in this editor is shown in Table 2, and an actual metadata database extract is shown in Table 3.

For this work, standardised metadata specifications, such as the Dublin Core and IMDI sets, were taken into account. However, new resource types such as those which are characteristic of linguistic fieldwork demonstrate that the standards are still very much under development, since some of the standard metadata types are not relevant for the fieldwork data, and the fieldwork data types contain information not usually specified in metadata sets, but which are common in the characterisation of spoken language resource databases (Gibbon et al., 1997). In respect of the fieldwork resource type, it appears that it cannot be expected that a truly universal — or at least consensual — set of corpus metadata specifications will be developed in

Table 3: Fieldwork metadata example.

Attribute	Value
RecordID:	Agni2002a
LANGname(s):	Agni, Anyi
SILcode:	ANY
Affiliation:	Kwa/Tano
Lect:	Indni
Country:	Côte d'Ivoire
ISO:	CI
Continent:	Africa
LangNote:	
SESSION:	FieldIndoor
SessionDate:	11.3.02
SessionTime:	8:57
SessionLocale:	Adaou
Domain:	Syntax
Genre:	Questionnaire
Part/Sex/Age:	Kouamé Ama Bié f 35
Interviewers:	Adouakou
Recordist:	Salffner, Gibbon
Media:	Laryngograph
Equipment:	1) Audio: 2 channels, l laryngograph, r Sennheiser studio mike 2) S tills: Sony digital 3) Video: Panasonic digital (illustration of techniques)
SessionNote:	Adouakou phrases repeat

the near future, or perhaps at all, at a significant level of granularity. It may be possible to constrain the attribute list, though the existence of many different fieldwork questionnaire types belies this. However, the values of the attributes are in general unpredictable, entailing not only free string types but possibly unpredictable rendering types (e.g. different alphabets; scanned signatures of approval).

The metadata specifications used in the UBIcorpus applications are deliberately opportunistic, in the sense that they are task-specific and freely extensible. A selection of attributes and values for the current fieldwork application are shown in Table 2. Metadata attributes concerned with the Resource Archive layer of archiving and property rights are omitted.

For current purposes, databases are exported in the attribute-value format shown below and converted into the TASX reference XML format (Milde and Gut, 2001). A specific example of the application of the metadata editor in the fieldwork session pictured in Figure 5 is shown in the exported record shown in Table 3.

7. Step 5: Corpus creation

The corpus creation process has been described in terms of three phases (Gibbon et al., 1997):

Pre-recording phase: planning of the overall corpus structure and contents, in particular design of corpus recording sessions, including the preparation of scenario descriptions, interview strategies, questionnaires, data prompts (for instance with prompt randomisation); most of what has been discussed in the preceding sections applies to this first phase. In systematic terms, the pre-recording phase involves two subphases:

1. Requirements specification (specification of needs),
2. Design (including the contents and structure of the corpus).

Recording phase: conduct of corpus recording sessions, including session management with the logging of metadata in a metadata editor and database, questionnaire consultation and data prompt presentation. This phase may also be thought of as an *implementation* phase. Note that playback equipment is also generally required, for checking quality or, in fieldwork situations, playing back to those whose voice was recorded. The choice of equipment is bewilderingly varied, but the main points to note in this context are:

1. Avoid storage techniques which employ lossy compression (e.g. MP3 format storage), since they distort signal properties which can be important for phonetic and psycholinguistic research such as spectral tilt, an important voice quality feature in emotional speech.
2. My own standard recording equipment includes:
   - portable DAT cassette tape recorder,
   - high quality microphones (I usually record in stereo),
   - high quality headphones,
   - battery driven loudspeaker for playback to an audience,
   - portable field laryngograph (for intonation and tone resource creation),
   - digital video recorder,
   - Palm handheld interview prompt and metadata log database,
   - requisites for experiments, such as coloured blocks,
   - paper and pen writing equipment,
   - plastic bags for storing equipment in the tropics,
   - lots of batteries.

Post-recording phase: provision of recorded and logged data for archive storage and processing, including metadata export, transcription, lexicon development, systematic sketch grammar support and document production; some of what has been discussed in the preceding sections applies to this third phase. Of course the entire corpus application procedure is also chronologically 'post-recording', but a line between resource development and application development has to be drawn, though it may sometimes be a little fuzzy.

8. Example 1: Corpora in language documentation

Fieldwork is a special case of language and speech corpus creation, by far the most complex case, concerned with providing data for specific documentary and descriptive purposes. Linguistic fieldwork is embedded in a complex network of intellectual, emotional, social, social and economic constraints. Some of the logistic factors in this environment are shown informally in Figure 1.

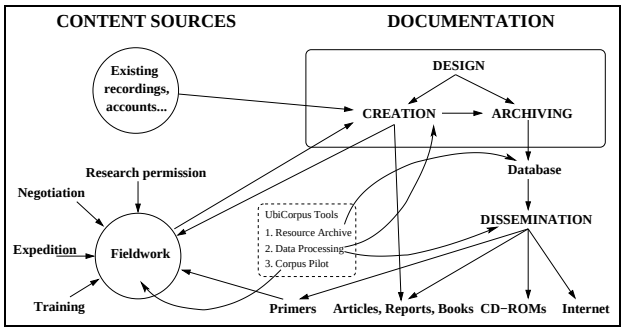


Figure 1: Logistics of language documentation.

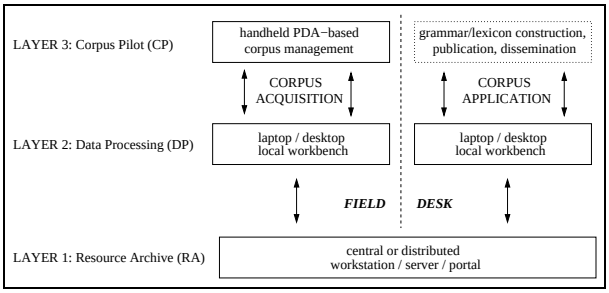


Figure 2: A flexible architecture for language documentation.

Figure 3: Scripted interview on Ega orature.

A more specific architecture for corpus processing in the field, using techniques described in previous sections, is shown in Figure 2. In the area of corpus management a PDA such as the Palm handheld, with an approprite database system, can be very useful, in particular in metadata planning and prompt creation at the design stage, but also in prompted elicitation and interview, and in metadata logging based on the design considerations. A further use is in lexical database construction, but also in design of computational corpus analysis strategies. The role of the PDA

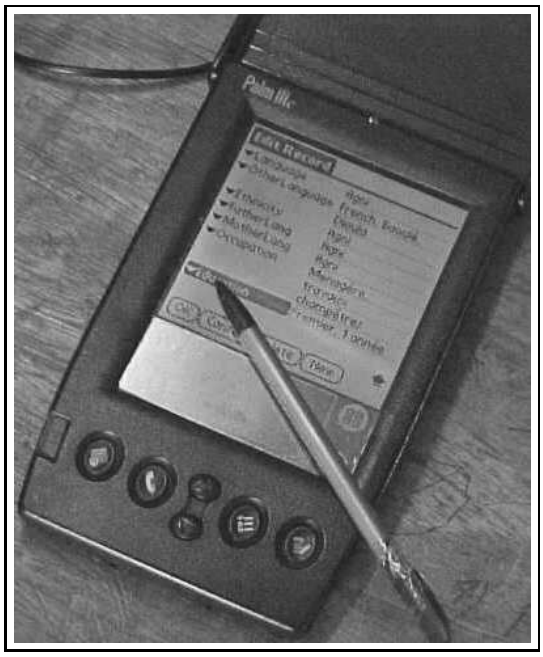


Figure 4: Palm handheld metadata editor.

Figure 5: Questionnaire interview on Anyi tone.

in resource construction is illustrated in a little more detail in Figure 2, in terms of a three-layer architecture describing the acquisition and the use of linguistic resources, with the resources in the bottom layer, standard desktop environments in the middle layer, and PDAs for mobile use in many field environments in the top layer.

The deployment of carefully planned questionnaires for systematic sociolinguistic and grammatical analysis work, and 'screenplays' for the elicitation and capture of interactive situations, are a central feature of fieldwork activity, and are eminently suitable for the handheld. Figure 3 shows a guided interview on oral narrative, with the narrator of the Ega village Nyigedugu (Gnieguedougou), Ivory Coast, facing. The interview was prepared by Sophie Salffner, a linguistics student training as a fieldworker, and recorded on digital audio tape. The metadata description had been prepared beforehand with the PDA, and the basic interview question text had been prepared on a desktop and transferred to the PDA for use in the field. The author is seen conducting the interview, using the PDA for reference.

Figures 4 and 5 show a different interview situation, in which Sandrine Adouakou, a doctoral student from Ivory Coast, and Sophie Salffner, trainee fieldworker, are eliciting audio data for the study of tone language prosody. The scene is in a small house in the Anyi Ndenye (Agni Indenie) village of Adaou, Eastern Ivory Coast, near the Ghana border. In this situation, the PDA is being used continuously to log metadata manually, since it was not possible to plan the elicitation procedure beforehand on this occasion, which arose at short notice.

9. Example 2: Securing interpretability of corpora

In a recent experiment, five types of legacy resource for an endangered language, including corpus data, were subjected to a process of interpretability securing (Gibbon et al., 2004). Interpretability securing means the preparation of a corpus in a sustainable format and with sufficient metadata to ensure that it can be decoded (archive interpretability) and understood (language interpretability) in the permanent absence of native speakers.

The five types of resource, for the Ega language which has been mentioned in preceding sections, are:

Lexicon:

Problem: a minimally documented Shoebox lexical database with a lexicon of the language; Shoebox is a toolkit or workbench for descriptive lexicography in fieldwork situations.

Solution: a conversion procedure for identifying fields in the database and converting the Shoebox format to a well-defined data structure in XML was developed. There were limits to the faithfulness of the procedure, since some fields were not fully documented.

Character encodings:

Problem: some data, including the Shoebox lexical database, used unspecified or unavailable proprietary fonts, resulting in renderings with uninterpretable symbols.

Solution: reference was made to PDF/print versions in reverse engineering the encoding, and characters were redefined in Unicode.

Interlinear glossed text:

Problem: very little glossed text was available, and in particular no account was included of the all-important tones, which in this language have both lexical and morphosyntactic functions (i.e. not only in distinguishing words, but also in tense marking).

Solution: phonetic tone annotations and the relevant lexical and grammatical categories for tone assignment were added in additional annotation tiers, based on a sketch grammar of the language.

Annotated recordings:

Problem: transcriptions and annotations have been made in many formats (Xwaves, Praat, Transcriber, ...), not all of which contain the same amount of information; some of the discrepancies are in the metadata types which are included, another discrepancy is that some formats associate transcription symbols with single time-stamps for temporal points, while others associated transcription symbols with dual time-stamps, for temporal intervals.

Solution: the different annotation formats were normalised to the TASX XML format, preserving a maximum amount of information; single time-stamp formats were normalised to dual time-stamp formats.

Linguistic descriptions:

Problem: various descriptions were available in different descriptive linguistic traditions; all were incomplete in different ways.

Solution: an attempt was made to use the General Ontology for Linguistic Descriptions (GOLD), recently developed in the EMELD project by Langendoen and Farrar as a normalised category set. However, both the resources had gaps in respect of the GOLD ontology, and the GOLD ontology had gaps in respect of the resources. This problem is easily the hardest, owing to the incomparability of theories with different premises and structures.

On the basis of these results, metadata definitions were provided for OLAC metadata repository; cf. Section 6. In the ideal case, the actual data referred to by the metadata would be stored in a *trusted repository* such as an institutionally based archive or library, rather than a departmental server.

10. Conclusion

If approached systematically, corpus building can be a reasonably straightforward task. In a laboratory situation, the procedures are rather easily controllable, but in a linguistic fieldwork situation this is not at all the case. Negotiation procedures - which have not been touched on here - and ethical issues - which have also not been touched on - loom large, and the local rhythms of everyday life exert strong constraints on the time available for corpus recording.

In a fieldwork situation, the slightest mishap may perturb the progress of work, or even prevent it. Some cases from my own experience...

1. Identical connectors on different low tension voltage sources were exchanged, which destroyed the power supply of a digital video camera, which was not repairable in the field or in local towns.

2. A device for airflow measurement arrived at the last minute and could not be checked; on opening in the field it turned out that one small but crucial tube was missing. Fortunately a medical supplier in a suburb of Abidjan had tubes of the same size in stock.

3. In an inattentive moment due to the tropical climate a student jammed a DAT tape into the DAT recorder wrong way round and could not extricate it. A more robust colleague was about to dig it out with a table knife, but I preferred to apply the methods described in (Pirsig, 1974), and sat down for an hour or so with a cup of tea meditating on the problem (and checking the delicate mechanism with a torch), finally removing the offending cassette undamaged by levering an almost invisible catch with a fine-bladed penknife. The more robust colleague then asked me why I hadn't done this right away instead of wasting time drinking tea...

For reasons such as these, extensive prior equipment testing, and a certain amount of redundancy of equipment is necessary. But in fieldwork corpus building there is inevitably an element of 'no risk, no fun'...

11. References

Bird, Steven and Mark Liberman, 2001. A formal framework for linguistic annotation. *Speech Communication*, (33 (1,2)):23–60.

Gibbon, Dafydd, Catherine Bow, Steven Bird, and Baden Hughes, 2004. Securing interpretability: the case of Ega language documentation. In *Proc. LREC2004*. Paris: European Language Resources Association.

Gibbon, Dafydd, Inge Mertins, and Roger K. Moore (eds.), 2000. *Handbook of Multimodal and Spoken Dialogue Systems, Resources, Terminology and Product Evaluation*. Boston/Dordrecht/London: Kluwer Academic Publishers.

Gibbon, Dafydd, Roger Moore, and Richard Winski (eds.), 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Gibbon, Dafydd and Thorsten Trippel, 2002. Annotation driven concordancing: the pax toolkit. In *Proceedings of LREC 2002*. Paris: ELRA/ELDA.

Milde, Jan-Torsten and Ulrike Gut, 2001. The TASX-engine: an XML-based corpus database for time aligned language data. In *Proceedings of the IRCS Workshop on Linguistic Databases*. Philadelphia: University of Pennsylvania.

Pirsig, Robert M., 1974. *Zen and the Art of Motorcycle Maintenance*. New York: William Morrow.

van Eynde, Frank and Dafydd Gibbon, 2000. *Lexicon Development for Speech and Language Processing*. Dordrecht: Kluwer Academic Publishers.

First steps in FP6

Bojan Petek

Interactive Systems Laboratory, University of Ljubljana
Snežniška 5, 1000 Ljubljana, Slovenia
Bojan.Petek@Uni-Lj.si

Abstract

This contribution presents a structured view of the ISCA SALTMIL SIG opportunities in the Sixth EU Framework Programme for Research and Technological Development (FP6). It overviews the SIG's early involvement in FP6 by describing a submission of the expression of interest (EoI) *HLTport* (*H*uman *L*anguage *T*echnology *Port*ability) to the programme, and summarizes reflections from participation on the "Information Day on the early stages of implementation of the IST programme within FP6" in Luxembourg. The *HLTport* FP6 EoI is one of the actions taken after the LREC 2002 workshop on "Portability Issues in Human Language Technologies". Building on the excellent FP6 presentation at CORDIS, the paper develops a closer look at the opportunities based on the intersection between the SALTMIL and the FP6 aims and objectives. The paper concludes by proposing ways aiming to help these opportunities to fulfill.

Final contribution

References

URL list accessed in April 2004:
http://www.isca-speech.org/
http://isl.ntf.uni-lj.si/SALTMIL/

http://fp6.cordis.lu/fp6/home.cfm
http://www.cordis.lu/fp6/fp6_glance.htm
http://www.cordis.lu/fp6/stepbystep/table_overview.htm

http://eoi.cordis.lu
http://eoi.cordis.lu/dsp_details.cfm?ID=32189
http://www.cordis.lu/fp6/eoi-analysis.htm
http://www.cordis.lu/ist/ka3/news.htm

http://www.lrec-conf.org/lrec2002/lrec/wksh/
Portability.html

http://isl.ntf.uni-lj.si/SALTMIL/lrec04/bp.pdf

Building an Amharic Lexicon from Parallel Texts

Atelach Alemu and Lars Asker
Department of Computer and Systems Sciences,
Stockholm University and the Royal Institute of Technology,
Forum 100, SE-164 40 Kista, Sweden,
email: [atelach | asker]@dsv.su.se
Gunnar Eriksson
Department of Linguistics, Stockholm University,
SE-106 91 Stockholm, Sweden
gunnar@ling.su.se

Abstract

We present an automatic approach to constructing a partial Amharic – English lexicon from parallel texts. The analysis of the texts is supported by the application of lemmatization and part of speech tags from a dependency parser for English and a simple morphological analyzer for Amharic. The results in this paper show how a small partial lexicon that maps a subset of English nouns to their correct Amharic counterpart, can be constructed with relatively good accuracy. We also describe a way of improving precision by providing a measure for the likelihood that a mapping is correct.

Introduction

Amharic is the official government language of Ethiopia and is spoken by 15 - 30 million people. In spite of this, it suffers severely from lack of computational linguistic resources. Most work on natural language processing of Amharic to date has consisted of limited prototypes of morphological analysers, part of speech taggers, and parsers that all suffered from the fact that no lexicon, or other resources existed and that they thus had to be constructed from scratch. The work presented in this paper deals with the construction of an English Amharic lexicon, where we try to benefit as much as possible from the existence of such resources for English.

We present an automatic approach to constructing an English-Amharic electronic lexicon from parallel texts. The approach builds on earlier work presented in (Alemu *et.al*, 2003), but is extended through the application of a simplistic Amharic morphological analyser that we implemented. In addition to this, we supplement the morphological analysis by part of speech tags for the English words. We have considered the possibilities to use a dependency parser (Tapanainen & Järvinen, 1997) for the English texts and the correspondences between parallel sentences to support the construction of the Amharic lexicon. A very strong claim of these correspondences is often referred to as the Direct Correspondence Assumption (DCA), which states that syntactic dependencies hold across languages. In its purest form this is certainly a too strong assumption, but as Hwa et al. (Hwa et.al, 2002) shows, some syntactic dependencies, such as head-modifier relations, often hold between parallel sentences in different languages.

In the work presented in this paper, we only use the lemmatized words and the Part-of-Speech provided by the dependency parser, to support the mapping between words in the two languages. The analysis is then further supported by doing a simplified morphological analysis of the Amharic words (i.e. stripping of the longest matching prefixes and suffixes).

It is obvious to the authors that the described approach can not be fully automatic and completely replace the work of human experts. Instead it is intended to function as a semi-automatic support to human experts, and to reduce their monotonous workload and instead allow them to focus their attention on the difficult linguistic considerations that have to be taken when building a lexicon.

Lexicon extraction from parallel texts

The work presented in this paper is based on the incorrect and oversimplified assumption that there is a one-to-one mapping between words in different languages. If this assumption would be correct, then for a given parallel text that is aligned at the sentence level, there would always exist a unique mapping between word pairs across parallel sentences. Furthermore, the words in each such word pair, would have an identical distribution over the whole parallel corpus (and over the language in general). A word in language A that occurs only in sentences S_{A1} … S_{An} would have a corresponding word in language E that occurs in the aligned sentences S_{E1} … S_{En} (and only in these sentences).

There are a lot of exceptions to this assumption that in reality makes the mapping between words a more difficult task. Words in one language will have a corresponding translation that consists of more than one word in the other language, they might not occur at all or as affixes or parts of words in the other language, they might be ambiguous in one language but not in the other, they might be expressed as a number of synonyms in one or both languages and not have a clear overlap, etc. etc.

In spite of this, we believe that this one-to-one mapping occurs often enough for us to take advantage of it in the automatic extraction of a bilingual lexicon.

The data set

We are currently working with English[1] and Amharic[2] versions of the New Testament. Due to the fact that these texts are numbered down to the verse level, they were relatively easy to align. The manual work consisted of adjusting for the few occurrences of different numbering in the two texts. These texts had the advantage that they consisted of a considerable amount of translated text (100,000 words) and that they were already relatively well aligned at the sentence level. The two data sets are available on the Internet in xml-format. Amharic uses its own and unique alphabet (Fidel) and there exist a number of fonts for this, but to date there is no standard for the language. The Amharic data set was originally represented using a Unicode compliant Ethiopic font called Jiret. For compatibility reasons we transliterated it into an ASCII representation using SERA[3].

Experiments

Due to the different morphological properties of the two languages (Amharic being highly inflected while English is not), we wanted to utilize the fact that a correct mapping between an English word and an Amharic word in the parallel text could be expressed in an number of different ways on the Amharic side and only a few on the English side. Given that we want to find a correct translation for an English noun, we would then start by locating all the English sentences that contain this word, and then look for the Amharic word that would occur in as many as possible of the corresponding Amharic sentences, and that would at the same time occur in as few as possible of the other sentences.

We calculated the information content in order to score the candidate words and to get a measure for the likelihood for each of the candidate words that they are the correct translation of the English word. The information content (which is a measure from information theory) finds the optimal number of bits that are required to code class membership.

Given an English word W_E that occurs in sentences S_{E1} .. S_{En} and only in those sentences in the corpus, W_E partitions the set of English sentences into two groups, one would be the group of sentences (G1) that contain W_E (S_{E1} .. S_{En}), and the other group would be the remaining English sentences (G2). It would also indirectly partition the set of Amharic sentences into two groups, those that are aligned with the sentences in G1 (S_{A1} .. S_{An}) and those that are aligned with the sentences in G2. If the English word W_E would always be translated into the same unique Amharic word W_A, and if W_E is the only English word that is translated into W_A, then W_A would partition the set of English and Amharic sentences into the same groups G1 and G2 that W_E does. Even in cases when a unique one to one mapping does not exist, we assume that the Amharic word would have a distribution among the Amharic sentences that is similar to the distribution for the English word. The task is therefore, for each Amharic word that occurs in at least one of the sentences S_{A1} .. S_{An}, to measure how well it can partition the sentences into the groups G1 and G2. We get a measure for this from the formula for information content given below:

$$\textit{Information Content for word } W_{An} =$$
$$-\left(C_{11} * \log 2\left(\frac{C_{11}}{C_{11}+C_{21}}\right) + C_{21} * \log 2\left(\frac{C_{21}}{C_{11}+C_{21}}\right)\right)$$
$$-\left(C_{12} * \log 2\left(\frac{C_{12}}{C_{12}+C_{22}}\right) + C_{22} * \log 2\left(\frac{C_{22}}{C_{12}+C_{22}}\right)\right)$$

where C_{11} is the number of aligned sentences that contain W_E and W_{An}, C_{12} is the number of aligned sentences that contain W_{An} but not W_E, C_{21} is the number of aligned sentences that contain W_E but not W_{An}, and C_{22} is the number of aligned sentences that does not contain neither of W_E or W_{An}.

Pertaining to the difference in the morphological properties of the two languages the mapping is fewer to more in most cases. If we take for example the Amharic word bEt which means house, there are many forms of this noun that could be mapped to house. But each of the word forms would appear as a separate word and would have a frequency count of its own.

bEt -- house

bEtu – the house

yebEtocE – my houses'

bEtacew – their house

kebEtu – from the house

yebEtum – the house's also

yebEtocachu – your houses'

lebEtocacn – for our houses

Amharic nouns are inflected with plural markers, possession markers, definiteness markers, object case markers and emphasis markers. Furthermore, prepositions cliticize to the first element of the noun phrase. Thus, a single Amharic noun such as bEt may have more than 100 inflected forms. In order to handle this, morphological analysis (here just stripping of the prefixes and affixes) of the Amharic words was done before the frequency counts.

To lemmatise and tag the English words the ENGFDG (Tapanainen & Järvinen, 1997), dependency parser has

[1] Parallel Corpus Project: The Bible, available at: http://benjamin.umd.edu/parallel/bible.html

[2] Revised Amharic Bible in XML, available at: http://www.nt-text.net/eth/eth_index.htm

[3] SERA stands for System for Ethiopic Representation in ASCII, http://www.abyssiniacybergateway.net/fidel/sera-faq.html

been used. The information about the word class of the English words was then used to guide the morphological analysis of the Amharic counterparts.

We have run a number of preliminary experiments where we, for each English word, first scored all the Amharic candidate terms for that word. A candidate term is a word that occurs in at least one of the Amharic sentences that represents translation of the English sentences in consideration. We then selected a subset of the English words. We assumed that it would be easier to come up with a correct mapping if the candidate terms would occur in more than just a few sentences and therefore selected a subset where the word in consideration had occurred in at least 7 different English sentences, and where the top ranked Amharic translation candidate had occurred in at least two different sentences. In addition to that we only considered English words that had been tagged as nouns. This resulted in a subset consisting of 545 distinct English nouns. For each such English word, we then selected the 30 highest translation candidates from the Amharic side for further processing.

Evaluation

We manually evaluated the 30 highest ranked translation candidates for each of the 545 English words. The evaluation consisted of determining what candidates were to be considered as correct translations of the English word, what were to be considered as synonyms to (or different word forms of the correct translation), and to determine in what cases the correct translation would consist of more than one word (e.g. church = bEte krstiyan).

In the first part of the experiments, where we used no morphological analysis of the Amharic words, we managed to correctly map 417 of the 545 (76.5%) of the words correctly, in 125 cases (23.5%) a wrong word was found to be the top ranking candidate. In 22 cases the correct Amharic translation was a term consisting of two or more words, and in all these cases, one of the consisting words were ranked as the top candidate. In 17 of those, all words were consecutively ranked as the top scoring ones, while in 5 cases, the second word was ranked as the 3rd or 4th highest ranking candidate.

In the second part of the experiments, we merged different forms of the Amharic candidate words by ignoring differences in inflection by prefixes and suffixes. Here, we managed to correctly map 458 of the 545 (84.0 %) of the words correctly, in 87 cases (16.0 %) a wrong word was found to be the top ranking candidate. In 22 cases the correct Amharic translation was a term consisting of two or more words. As above, in 17 of those cases, all words were consecutively ranked as the top scoring candidates, while in 5 cases, one of the words was ranked as the 1st candidate, but the second was ranked as the 3rd or 4th highest ranking candidate. These results are presented in Figure 1, below. “correct” is the number of words that were mapped to their correct translation, “synonym” is the number of words that were mapped to a synonym, “two words” are the number of words that were correct Amharic translations consisting of two or more words and where these were the top ranking candidates, and “incorrect” is the number of words that were incorrectly mapped.

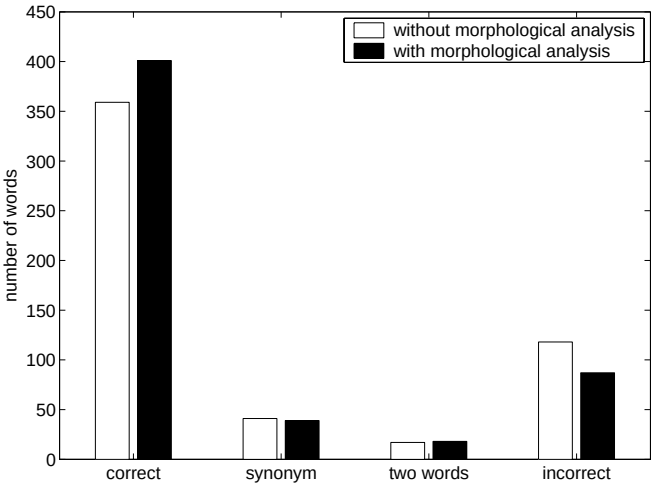


Figure 1. Results from the experiments for the not morphologically processed and the morphologically processed datasets, respectively.

We noted an interesting difference between the cases when the English word was mapped to its correct Amharic translation, and the cases when the mapping was not successful. In most cases the mapping succeeded, there was a relatively large difference between the score for the highest ranked (correct) Amharic word, and the remaining translation candidates. On the other hand, when the algorithm did not rank the correct translation as the top ranking alternative, this difference was in most cases much smaller.

Since in these experiments, we are not primarily trying to generate a complete lexicon, but rather a smaller partial lexicon, we are interested in finding an estimate of how likely it is that a certain mapping between an English and an Amharic word is correct. We therefore filtered the result by ignoring such suggested mappings that have an absolute value of the difference of the score for the highest and second highest ranked candidates below a certain threshold. The assumption here would be that a lager difference would more clearly indicate that the mapping is correct. In Figure 2 below, we show the results of this filtering process (on the morphologically processed words) as a tradeoff between precision and recall for the English words as a function of the threshold value. The precision is measured by dividing the number of correctly mapped word pairs by the number of correctly mapped word pairs plus the number of incorrectly mapped word pairs. The recall is measured by dividing the number of correctly mapped word pairs by the total number of English words under consideration. As can be seen from the figure, as threshold values approach zero, the precision and recall would get closer and closer to 84% and 100% respectively, which are the same figures that are reached when no filtering is applied. On the other hand, as the threshold values increase, the precision will

increase while the recall will decrease, so that for example at a threshold value of 10, the algorithm will have a precision of 95% and a recall of 80% of the words.

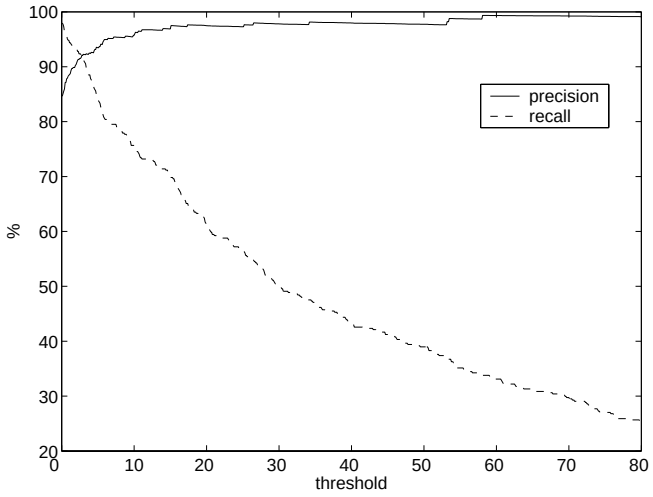


Figure 2. Precision and recall for the morphologically processed words, as a function of threshold value when morphological analysis has been done.

Conclusions

We have shown preliminary results of experiments aimed at finding automatic ways to create a partial English – Amharic lexicon. For a limited subset of words that occur frequently enough in the corpus, we have shown that it is possible to extract correct mappings between English and Amharic words with a relatively good precision.

Since the amount of words used is very small and the experiments are conducted on a single word class, it is difficult to draw major conclusions. Nonetheless, simple morphological analysis seems to have a positive influence in automatic bilingual lexicon extraction. This could be accounted to the fact that the number of count for the corresponding Amharic words is much higher when the affixes are stripped off, and a better matching is obtained. Furthermore, our experiments suggest that the difference in scoring between the 1st and 2nd candidate word may be a good indicator of the reliability of mapping.

We have currently not dealt with the problem of determining when the correct mapping involves terms with more than one word in either language, but the few examples of such occurrences in the current experiment indicate that the difference in score between candidates can be an indicator of this. This will however have to be investigated in more detail before any definite conclusions can be made.

Future work

We are currently discussing different ways in which the current work can be extended. One way is to build up a larger lexicon iteratively by first creating a small and accurate partial lexicon, and then to make those mappings permanent in order to reduce the possible mappings for the remaining words. Another is to apply the same technique to do morphological analysis for word classes other than nouns. There is also the possibility to use a number of other heuristics to guide the mapping between words. These include string matching for proper names (which tend to be similar across languages). It also includes the application of linguistic knowledge about specific properties of Amharic (e.g. that the verb always comes last in a sentence). In the current work, we used an Amharic corpus consisting of approximately 100,000 words. It is reasonable to believe that the techniques described here will benefit from even larger data sets and we are actively looking for a larger parallel corpus in order to verify this.

References

Alemu, A., Asker, L., and Eriksson, G. “An Empirical Approach to building an Amharic Treebank”, In Proceedings of the Second Workshop on Tree banks and Linguistic theories (TLT 2003), 2003, Växjö, Sweden.

Tapanainen, P. and Järvinen, P., “A non-projective dependency parser”, In Proceedings of the 5th Conference on Applied Natural Language Processing, pages 6471, Washington D.C., April 1997. Association for Computational Linguistics.

Hwa, R., Resnik, P., Weinberg, A., and Kolak, O., ”Evaluating Translational Correspondence using Annotation Projection” , 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02), Philadelphia, July, 2002.

Spoken Language Corpora in South Africa

Jens Allwood
Department of Linguistics, University of Gothenburg, Renströmsgatan 6, S-412 55 Gothenburg, Sweden
Visiting scholar: Department of Linguistics, Unisa, jens@ling.gu.se
A. P. Hendrikse
Department of Linguistics, University of South Africa, P. O. Box 393, Pretoria 0003, South Africa, hendrap@unisa.ac.za

Abstract

In this paper we give an outline of a corpus planning project which aims to develop linguistic resources for the nine official African languages of South Africa in the form of corpora, more specifically spoken language corpora. In the course of the article, we will address issues such as spoken language vs. written language, register vs. activity and normative vs. non-normative approaches to corpus planning. We then give an outline of the design of a spoken language corpus for the nine official African languages of South Africa. We consider issues such as representativity and sampling (urban-rural, dialects, gender, social class and activities), transcription standards and conventions as well as the problems emanating from widespread loans and code switching and other forms of language mix characteristic of spoken language. Finally, we summarise the status of the project at present and plans for the future.

Introduction

In this article we give an outline of a joint corpus linguistics project between the Departments of Linguistics at Unisa and Gothenburg (Sweden). The project aims to develop computer-based linguistic resources for the nine official African languages of South Africa in the form of spoken language corpora. The raw data of the corpora come from audio-visual recordings of natural language used in various social activities.

Although this project is administered by the two linguistics departments mentioned above, we would like to involve as many African linguists and scholars as possible working on these languages as full participants in this project. One of the aims of this article, then, is to publicise this project, its goals, methods and potential outcomes to the relevant community of scholars in South Africa.

The rationale behind the project

Diminished and diminishing linguistic diversity is a characteristic feature of our contemporary world. This feature is, to a large extent, a function of the effects of globalisation on diversity. Factors such as global socio-economic pressures, the need for international communication standards and stable geo-political relations seem to entail inevitable monolingualism at the expense of linguistic diversity. About half of the approximately 6 000 languages spoken in the world today will be extinct by the end of the century for the simple reason that 90% of the world's population speaks the 100 most-used languages (Nettle & Romaine, 2000: 8). Even some of the 100 most-used languages may ultimately succumb to what Granville Price (as quoted by Nettle & Romaine, 2000:5) has aptly called the "killer language", namely English or, more precisely, World Englishes. English in all its varieties is simply the predominant medium of international linguistic interaction.

Why, then, given these overwhelming trends towards global monolingualism, should any speech community channel any efforts and resources towards the maintenance of their language? In a sense, the Asmara Declaration, which was issued by the delegates to a conference entitled *Against All Odds: African Languages and Literatures into the 21st Century* held in Asmara, Eritrea from 11 – 17 January 2000, is an attempt to answer this question.

1. The vitality and equality of African languages must be recognized as a basis for the future empowerment of African peoples.
2. The diversity of African languages reflects the rich cultural heritage of Africa and must be used as an instrument of African unity.
3. Dialogue among African languages is essential: African languages must use the instrument of translation to advance communication among all people, including the disabled.
4. All African children have the inalienable right to attend school and learn in their mother tongues. Every effort should be made to develop African languages at all levels of education.
5. Promoting research on African languages is vital for their development, while the advancement of African research and documentation will be best served by the use of African languages.
6. The effective and rapid development of science and technology in Africa depends on the use of African languages, and modern technology must be used for the development of African languages.
7. Democracy is essential for the equal development of African languages and African languages are vital for the development of democracy based on equality and social justice.
8. African languages, like all languages, contain gender bias. The role of African languages development must overcome this gender bias and achieve gender equality.
9. African languages are essential for the decolonization of African minds and for the African Renaissance.

The South African spoken language corpus (SASLC) project subsumes, directly or indirectly, all the concerns expressed in this declaration, but more specifically the concerns raised in points 3 – 6, in the sense that it will develop a platform of computer supported basic linguistic resources for applications in translation (point 3), language teaching (point 4), language development (point

5) and language adaptations for science and technology (point 6).

Compared to corpora of English, SASLC is perhaps most similar to the Wellington corpus of spoken New Zealand English (Holmes et al., 1998), to the spoken language part of BNC (British National Corpus) and to the London/Lund corpus (Svartvik, 1990). Compared to spoken corpora of the Nordic languages, SASLC is similar to the Danish BySoc corpus (Gregersen 1991; Henrichsen 1997). The SASLC project is however distinct from these spoken language corpora in that its sampling is activity related, i.e. natural language use in a representative range of socio-economic activities. In this regard, SASLC is very similar to and largely guided by the approach of the Gothenburg spoken language corpus (GSLC). See Allwood et al, 2001)

In the next section we briefly contrast spoken and written language and indicate why we focus on spoken language in this project.

Why spoken language?

Structuralist linguistics for a long time has favoured (explicitly and perhaps mostly implicitly) the view that the difference between spoken and written language is of no relevance to linguistic theory. In addition to the more applied objectives of the SASLC project (such as language development) we also aim at a critical examination of this linguistic orthodoxy.

One reason for this is that the structure of spoken and written language, although similar in some respects, are also very different in many ways. Face-to-face spoken language is interactive (in its most basic form), multimodal (at the very least containing gestures and utterances) and it is also highly context-dependent. Further, spoken discourse very often consists of one word utterances. Written language, on the other hand, in its most typical form is non-interactive, monological and monomodal with a lesser degree of contextualisation. Typically, written language involves sentences which are governed by normative rules that dictate the structure of properly formed sentences. The norms of spoken language are usually of a different sort, rather dictating communicative efficiency enabling high rate processing required by speech.

In spoken language we therefore find linguistic expressions that enable "online" thought processing or expressions that allow for change of mind. From a normative written language perspective these linguistic phenomena might be called "dysfluencies", "false starts", "self-corrections" etc. In spoken language one also finds short and unobtrusive ways of giving discourse feedback, e.g. expressions like *ee, mh*, *yuh* that indicate comprehension, affirmation, surprise and so on.

None of these linguistic phenomena that are so characteristic of spoken language have any place in written language. Through the development of spoken language corpora we therefore hope to broaden the empirical basis for work on what we believe ought to be the central area of linguistic research, namely face-to-face linguistic interaction.

Considerations in the compilation of a spoken corpus

The compilation of a spoken corpus in the multilingual environment in South Africa is seriously affected by at least two features of everyday language use: dialectical variations, on the one hand and, on the other hand, interlingual communicative strategies, such as loans, code-switching, urban koines (cf. Schuring, 1985). If one is aiming at recording natural language use, as we are, all the natural features of language use in a multilingual society, including dialectal variation and language mix, need to be recorded and accounted for.

Obviously, representativity depends on the kinds of variables that are selected to guide the empirical scope of the study. The deliberate bias of our project is on language use in a representative sample of social activities. This does not mean that we ignore other equally important variables. We deal with these variables in a particular annotational fashion rather than using them in the sampling criteria. In the SASLC representativity does not allude to sociolinguistic variables such as regional dialect, gender, social class or age but rather the range of social activities. We do this because we want to get an ecologically valid picture of the functionality of a language, which would be very difficult to achieve were we to use the traditional interview format which is normally used to capture variation with regard to regional dialect, gender, social class or age.

The project

Four initial phases are distinguished in the project: a recording phase, a transcription phase, a checking phase and a tagging phase. The overall progress of the project involves concurrent research activities in all four phases. In fact, the developments of various corpus tools, the creation and refinement of an archiving infrastructure, the training of research participants and even trial runs of research outputs require collateral work in all the phases more or less simultaneously.

The recording phase

This phase in the design and development of a corpus presupposes certain fundamental assumptions about various aspects of the data that will form the corpus. Generally speaking, the following parameters seem to guide such assumptions:

- representativity of the corpus
- control of variables in language varieties
- recording medium and storage
- volume/size of the corpus
- length of each sample

Allwood (2001) gives an outline of a different basis for a representativity measure for spoken language corpora, namely social activities. Social activities have been taken as the basis for decisions on the range and scope of representative samples in the Gothenburg spoken language corpus of Swedish (GSLC).

In our pilot study on Xhosa we have recorded samples of activities such as meetings, teacher discussions and seminars, student discussion classes, sermons, burial services, kin group meetings, informal discussions and patient interviews in hospitals. The corpus size we are aiming at is a million words, which corresponds to roughly 200 hours of recordings, per language.

The transcription phase

There are two facets to the transcription of recorded samples in our project:

- meta-transcription information (the header)
- the transcription of the contributions of all the speakers
- in an activity with some mark-up (the body).

The meta-transcription information

The transcription that can perhaps best be described by means of an example.

Transcription Header
@ Recorded activity ID: V010501
@ Activity type: Informal conversation
@ Recorded activity title: Getting to know each other
@ Recorded activity date: 20020725
@ Recorder: Britta Zawada
@ Participant: A = F2 (Lunga)
@ Participant: B = F1 (Bukiwe)
@ Transcriber: Mvuyisi Siwisa
@ Transcription date: 20020805
@ Checker: Ncedile Saule
@ Checking date: 20020912
@ Anonymised: No
@ Activity Medium: face-to-face
@ Activity duration: 00:44:30
@ Other time coding: Various subsections in the activity
@ Tape: V0105
@ Section: Family affairs
@ Section: Crime
@ Section: Unemployment
@ Section: Closing
@ Comment: Open ended conversation between two adult female speech therapy students Bukiwe and Lunga at Medunsa.

Each information line is marked by the @ sign. The information lines with the exception of a few are self-explanatory and need no further comment. The information in the recorded activity ID line: V010501 specifies the following: V = Video, 01 = project number, i.e. the current spoken language corpus project, 05 = the number of the tape within this project. Each participant in a recorded activity in the project gets a unique code. That is F1 (where F = female) is uniquely associated with Bukiwe and will again be used if she participates in another recorded activity. The general rule is that participants in the transcription remain anonymous and that all information that could identify them is removed from the transcription and retained in a separate file that is not publicly available. Headers are open-ended information structures and additional information about the participants (for instance their age, level of education, knowledge of other languages) could be freely appended.

The transcription (the body)

The mark-up conventions used in the annotations of the transcriptions of recorded activities in this project follow the transcription standards developed in the Department of Linguistics at Gothenburg University (cf. Nivre no date).

Three types of lines are distinguished in the transcription body – a contribution line preceded by the dollar sign $ (for speaker), a comment line preceded by the @ sign where comments about certain peculiarities in a contribution are provided, and a section line indicated by the § sign where the subsections of a sample text are designated. Consider the example below.

§ At office
Section line
$A: uyakhonza kanene < > Contribution
@ < nod >

The section in the sample from which this excerpt comes is 'at the office'. Participant A makes the contribution *uyakhonza kanene*. While A is making this contribution she nods and this concurrent gesture is marked by the angle brackets < > and commented on in the comment line <nod>.

Elisions, overlaps, comments, pauses, lengthening

Next, we exemplify some more features, typical of spoken language, which are part of the transcription standard.

§ Religion
$B: uyakhonza kanene
$A: ndiyakhonza owu ndiyamthand{a} [4 < uthixo > ndiyamthanda andisoze ndimlahle undibonisile ukuba mkhulu nantso ke into efunekayo qha]4 kuphela
$B: [4 nantso ke sisi // e: e:]4
@ < personal name: God >

In the contribution of A, the curly brackets in *ndiyamthand{a}* indicate that written language would require *-a*. Typical of certain spoken language activities is the occurrence of overlaps where some participant(s) say(s) something during the contribution of the participant who has the turn. These overlaps are indicated by means of square brackets (and are numbered because there could be several) in the contributions of the participants whose turn it is. In the excerpt above the bracketed overlap 4 illustrates this convention. Comment information can be of several kinds, for example, gestures, loans, code-switching and also names. Pausing is indicated by *//* (slashes) and vowel length by *:* (colon).

The checking phase

The checking involves viewing a copy of the video recording while following the transcription. In our pilot study so far we have tried to arrange a meeting after each checking phase where the transcriber and the checkers discuss flaws in the transcription and try to resolve differences of opinion. The checking phase is not only important to ensure the reliability and validity of the corpus, but also functions as a feedback to recorders to improve recording techniques.

The tagging phase

We will now briefly comment on the development of a tag set for African languages in our project. The

extensive inflectional variety within categories (e.g. up to 23 different classes of nouns with equally extensive concomitant concordial agreement varieties) requires some decision on the scope of the tag set. Should it represent slots/types and leave the paradigmatic varieties/tokens unspecified. For example, should the tag set only represent word classes, say, Noun without further reflection of the category- internal class distinctions, or should it represent the whole range of classes by means of different tags. We have opted for the latter approach in the development of a tag set in our project whereby paradigmatic varieties within a category are differentiated by means of different tags. Needless to say, this resulted in a rather sizeable tag set with rather serious implications for the manual tagging of the samples in the corpus.

The latter problem is addressed in several ways in the project. The tag set has been printed on charts (A1 paper size) in order to facilitate look-up. We are also in the process of developing computer-assisted manual tagging in the form of drag-and-drop tagging from tag set windows. And finally, we are currently developing an automatic computer tagger. Manual tagging is, however, still needed for the development of a training corpus and also for the correction of errors.

Conclusion

In conclusion we would like to briefly outline the scope of the potential research output of the corpus resources that will be developed in this project. Although the project is to some extent still in its beginnings stages where most activities were geared towards the building of an infrastructure as well as the training of researchers in the various facets of the project, sufficient progress has been made in some of our pilot studies to warrant the initiation of some research output activities as well.

Some of the possible long term results we hope to achieve through the project are the following:

(i) A database consisting of corpora based on spoken language from different social activities for the indigenous languages of South Africa. This database will be open to the research community, providing a resource for research and practical applications based on African languages.

(ii) A set of computer based tools for searching, browsing and analyzing the corpus. These tools will be developed in collaboration with the Department of Linguistics, Gothenburg University, Sweden.

(iii) Frequency dictionaries on the word level for the spoken language of the indigenous languages of South Africa. If written language corpora can be secured for these languages, we also expect to be able to provide comparative frequency dictionaries of spoken and written language for the same languages.

(iv) Frequency dictionaries based on morphological analysis of words.

(v) Analyses of a range of spoken language phenomena, such as own communication management and interactive communication (feedback, turn taking and sequencing).

(vi) Frequency based dictionaries for collocations and set phrases.

(vii) Descriptions of the language of different social activities, including, if this is seen as appropriate, frequency listings of words and phrases.

(viii) Syntactic analysis of spoken language and contributions to providing spoken language grammars for different African languages.

(ix) Analyses of spoken language, providing bridges to cultural analysis of narratives, values, politeness, etc.

These are nine possibilities we see at present. Which of them will actually be carried out will depend on the interests of the research team. Probably, as our work develops, also other types of analysis will appear.

Finally, let us reiterare the use that our corpora can have for comparative linguistic studies of African languages and for comparisons of non-African languages with African languages. In such comparisons, we hope to examine some typical spoken language phenomena such as feedback in comparisons between, for example, African languages, Afrikaans, English and Swedish.

The corpus can also be used as a resource for researchers and practitioners outside of linguistics, such as educators and speech therapists, for whom the corpus can serve as a basis for educational or therapeutic material or as an aid to the standardization of evaluative or diagnostic tests.22

References

Allwood, J., Grönqvist, L., Ahlsén, E. & Gunnarson, M. (2001.) Annotations and Tools for an Activity Based Spoken Language Corpus. In Kuppevelt J. (ed.), *Current and New Directions in Discourse and Dialogue*. Kluwer: Academic Publishers.

Allwood, J. (2001). Capturing differences between social activities in spoken language. In Kenesei, I. and Harnish, R. M. (eds.) *Perspectives on Semantics, Pragmatics and Discourse.* Amsterdam, John Benjamins, pp 301 –319.

Gregersen, F. (1991). *The Copenhagen Study in Urban Sociolinguistics, 1 & 2.* Copenhagen: Reitzel.

Henrichsen, P. J. (1997). Talesprog med Ansigtsløftning, IAAS, Univ. of Copenhagen. *Instrumentalis 10/97.*

Holmes, J. Vine, B. & Johnson, G. (1998). *Guide to the Wellington Corpus of Spoken New Zealand English*. Wellington: Victoria University of Wellington.

Nettle, D & Romaine, S. (2000). *Vanishing Voices: The Extinction of the World's Languages.* Oxford: Oxford University Press.

Nivre, J. No date. *Transcription Standards: Semantics and Spoken Language.* Göteborg University.

Schuring, G.K. (1985). *Kosmopolitiese omgangstale: Die aard, oorsprong en funksies van Pretoria-Sotho en ander koine-tale.* Pretoria: Raad vir Geesteswetenskaplike Navorsing.

Svartvik, J. (ed.) (1990). The London Corpus of Spoken English: Description and Research. *Lund Studies in English* 82. Lund University Press.

Montage: Leveraging advances in grammar engineering, linguistic ontologies, and mark-up for the documentation of underdescribed languages

Emily Bender*, Dan Flickinger†, Jeff Good‡, Ivan Sag†

*University of Washington
Department of Linguistics, Box 354350, Seattle, WA 98195-4340, USA
ebender@u.washington.edu

†Stanford University
CSLI/Ventura Hall, Stanford, CA 94305-2150, USA
{danf, sag}@csli.stanford.edu

‡University of Pittsburgh
Department of Linguistics, 2816 Cathedral of Learning, Pittsburgh, PA 15260, USA
jcgood@pitt.edu

Abstract

The Montage project aims to develop a suite of software tools which will assist field linguists in organizing and analyzing the data they collect while at the same time producing resources which are easily discoverable and accessible to the community at large. Because we believe that corpus methods, descriptive analysis, and implemented formal grammars can all inform each other, our suite of software tools will provide support for all three activities in an interoperable manner.

1. Introduction

The Montage (Markup for ONTological Annotation and Grammar Engineering) project aims to develop a suite of software whose primary audience is field linguists working on underdocumented languages. The tool suite is designed to have five major components: a manual markup tool to allow for basic grammatical annotation of data, a grammar export tool to allow annotated data to be summarized in a way similar to a traditional grammatical description, a labeled bracketing tool for incorporating information about syntactic relations into the data, a "grammar matrix" to assist with development of a precision formal grammar, and a tool which uses manually annotated data and a formal grammar to partially automate the annotation process.

2. Goal of the paper

The goal of this paper is to given on overview of the structure of the Montage toolkit with an emphasis on how it fits into the traditional conception of field work and language documentation and how the tools to be developed build off of existing tools for formal grammar engineering. Section 3 discusses which aspects of language documentation will be enhanced by the Montage toolkit. Section 4 describes the structure of the toolkit from a technical perspective. Section 5 describes how some of the tools which form the core of Montage will be adapted from existing tools for formal grammar engineering.

3. Language documentation and the Montage toolkit

Traditionally, the process of language documentation has been an extraordinarily labor-intensive and time-consuming task. It involves hundreds of hours of elicitation with native speakers. Based on such elicitation, basic documentary resources like audio and video tapes as well as annotated resources like transcribed texts and word lists can then be produced. After this is done, the data collector can begin to perform grammatical analysis on the language. In the ideal case, this work results in the creation of a descriptive grammar, a dictionary, and a small collection of translated and analyzed texts. Often, however, the barriers to the production of these documents are so high that they are never completed. When this is the case, the data from the language typically remains highly inaccessible and is effectively "lost" to the general community.

In the past few years, a small number of organizations have begun the project of developing digital standards and tools in order to make the task of language documentation easier as well as to ensure that digital resources created by field linguists are accessible to a wide audience and will not be lost as digital technology evolves. Some of these initiatives include the Electronic Metastructure for Endangered Languages project[1] (EMELD), the Dokumentation Bedrohter Sprachen project[2] (DoBeS), the Querying Linguistic Databases project[3], and the Open Language Archives Community[4] (OLAC).

These initiatives are developing tools that address some of the issues faced by field linguists. For example, the creation of dictionaries will be facilitated by EMELD's Field Input Environment for Linguistic Data tool[5] (FIELD), and the Elan tool[6], developed by the DoBeS project, is useful for the basic task of transcribing data. In addition to

[1] http://www.emeld.org
[2] http://www.mpi.nl/DOBES/
[3] https://www.fastlane.nsf.gov/servlet/showaward?award=0317826
[4] http://www.language-archives.org
[5] http://saussure.linguistlist.org/cfdocs/emeld/tools/fieldinput.cfm
[6] http://www.mpi.nl/tools/elan.html

these tools, the EMELD project is also developing an ontology of grammatical terms, called the General Ontology for Linguistic Description (GOLD) (Farrar and Langendoen, 2003). This ontology is designed to improve access to digital resources by creating a uniform means of annotating them for grammatical information, without necessarily imposing any particular theory or terminology on researchers.

Such tools represent an enormous change in the software available to field linguists. However, there remains a notable gap: Nothing yet available or currently under development supports the descriptive grammar component of field linguistic research.[7] The Montage toolkit will assist in such grammatical analysis, from foundational descriptive work to the statement and testing of precise hypotheses about grammatical structure.

Figure 1 illustrates which aspects of language documentation Montage is intended to facilitate. For illustrative purposes, the figure includes how two other tools—Elan and FIELD, discussed above—fit into this model of documentation. As schematized in the figure, Montage will (i) assist in creating annotated texts, specifically texts annotated for grammatical information, (ii) include tools for extracting information from the annotated texts to facilitate production of descriptive grammars, and (iii) allow information in descriptive grammars and electronic lexicons to serve as the foundation for the construction of formal grammars. As will be discussed in the next section, such formal grammars will be used by the system to partially automate annotation and analysis of data.

An important feature of Montage will be that it will allow grammatical annotations to be linked to external ontologies for grammatical terms. The use of ontologies will not be enforced in the toolkit, and the researcher will always have the freedom to use their own terminology. However, should they choose to use the terminology provided by the ontology or use other terminology but link it into the ontology, Montage will make this straightforward. The toolkit will, thus, be able to make important contributions to the creation of interoperable linguistic resources. The particular ontology which will be employed during the development of Montage is the GOLD ontology. However, the tookit's design will not restrict the user to any one particular ontology.

While implemented formal grammars have not traditionally been a part of language documentation, we believe that the current state of the art in computational linguistics is such that field linguists can now benefit from the enhanced hypothesis testing of grammar implementation without needing to become expert in a second subspecialty. Because of this, implemented formal grammars have an important position in Figure 1 with respect to the design of Montage—even if they don't fit into the traditional model.

In addition, we expect that the formal grammars produced by the toolkit will be valuable to software engineers working on tools which require knowledge of a language's grammar. To this point, such tools have generally only been available for majority languages. Montage will facilitate the creation of such tools for minority languages.

[7]The SIL tool, the Linguist's Shoebox, which has been in use for over a decade, can allow a linguist to perform basic text markup and, therefore, assist in grammatical analysis. However, this tool does not provide the support for the development of descriptive and formal grammars that is part of the design of Montage.

4. The design of Montage

The Montage toolkit will comprise five different tools, each of which could be used independently but which, when used together, will be designed to greatly enhance the workflow of the field linguist. The five tools are each discussed in turn.

- **Manual markup tool:** This tool will allow the markup of basic linguistic data for grammatical information. Its design will allow it to interface with an ontology of grammatical terms as well as with electronic lexicons so that morphemes in the data can be associated with their lexical entries.

- **Grammar export tool:** This tool will be a type of "smart" export tool to allow data annotated for grammatical information to be put into a format which facilitates traditional grammatical description. For example, it will export interlinearized example sentences as well as grammatial "notes" made by the linguist for particular linguistic constructions. Support will be included for creating both hyper-text grammars and traditional print grammars.

- **Labeled bracketing tool:** This tool will be similar to the markup tool except it will be specifically designed to annotate sentences for the phrase structure and to give grammatical labels to various levels of phrase structure. This tool will, therefore, facilitate syntactic description as well as the formation of formal implemented grammars.

- **Grammar matrix:** The Grammar Matrix is a language-independent core grammar designed to facilitate the rapid development of implemented precision grammars for diverse languages. (It will be discussed in more detail in section 5)

- **LKB/[incr tsdb()] tools:** These are two existing tools, the Linguistic Knowledge Builder and the [incr tsdb()] Competence and Performance Laboratory which will be used together to allow for semi-automatic parsing of data to find candidate sentences for possible grammatical annotation. (These tools will be discussed in more detail in section 5)

Figure 2 schematizes the workflow of grammatical description using the Montage toolkit. An important aspect of workflow using Montage is the "positive feedback loop" seen in the diagram. After the researcher manually marks up a set of data and creates a partial formal grammar, Montage will examine an entire corpus to find sentences not annotated for a particular grammatical feature but which would be good candidates for such annotation. A partially annotated corpus can, therefore, "jump-start" the process of annotating an entire corpus.

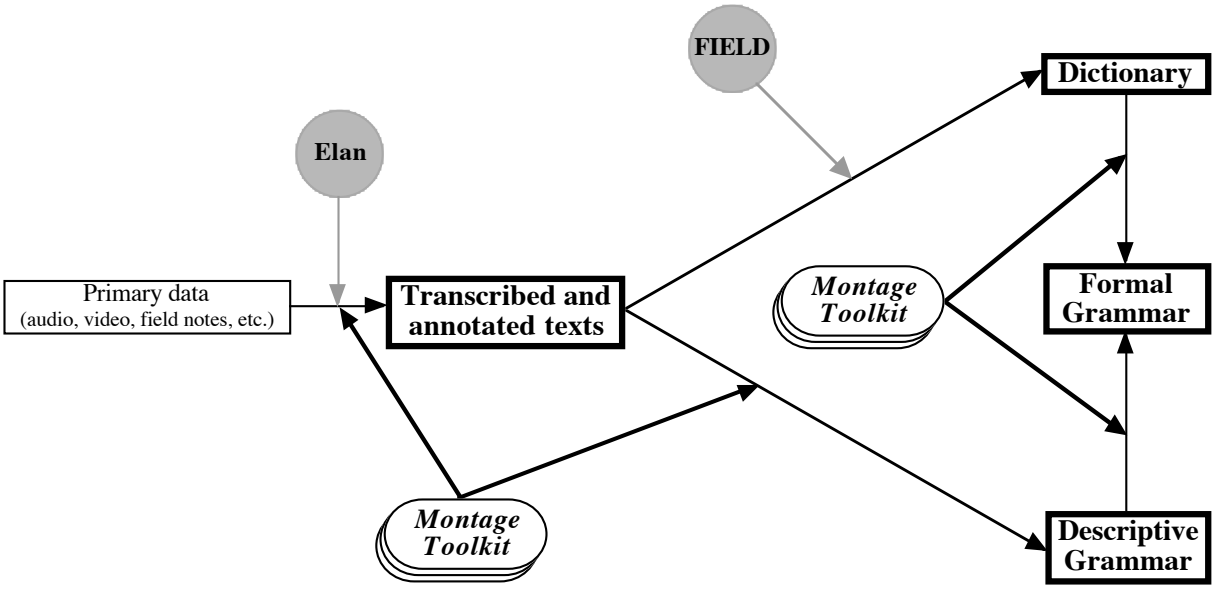


Figure 1: Language documentation and the Montage toolkit

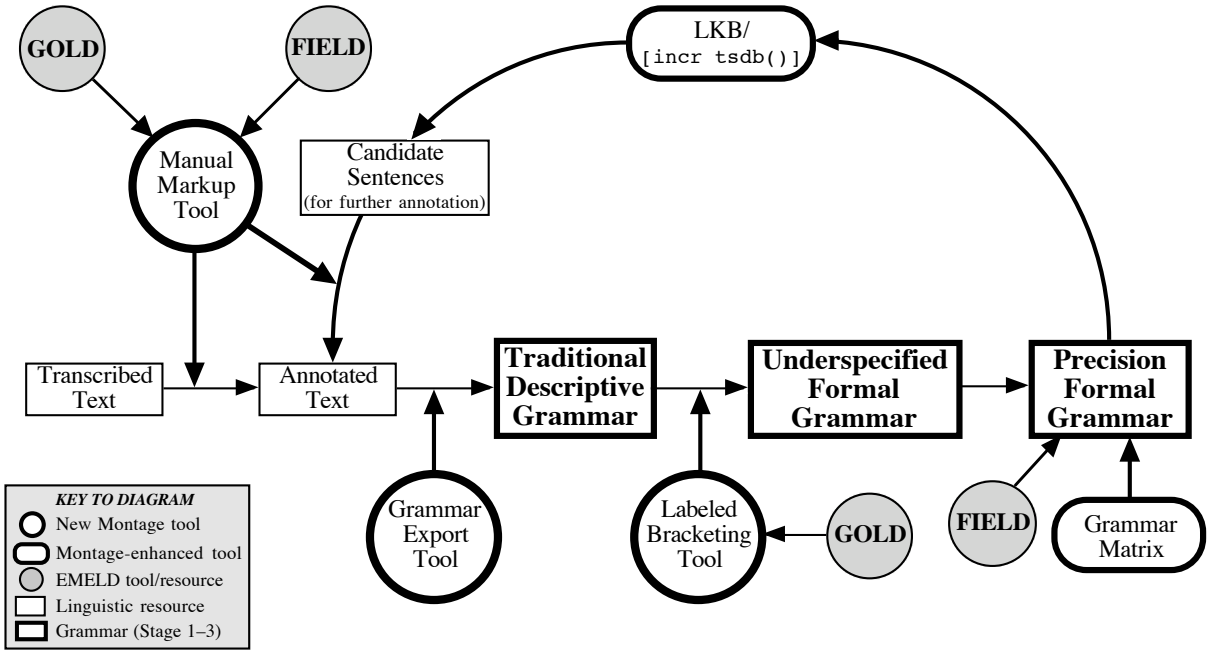


Figure 2: Workflow using the Montage toolkit

As should be clear from Figure 2, the Montage toolkit does not assume that work on the grammar of a language proceeds "serially". Rather, it assumes that work on each resource can assist in work on the other resources. For example, a partial descriptive grammar can assist in the production of a partial formal grammar which, in turn, can assist in the annotation of texts.

This model is designed with two ideas in mind. The first is that traditional field work largely proceeds in this "parallel" fashion—for example, informal work on grammatical description typically accompanies text analysis with no strict division of the work. The second reason for this model of workflow is to ensure that even incomplete grammatical analysis can produce a range of valuable resources. A partially annotated corpus of texts can easily be produced along side of a partial descriptive or formal grammar, for example. This will allow researchers to collaborate more easily on grammatical description and, crucially, will not necessitate that grammatical analysis only be publicly disseminated after it is "complete".

5. Refining existing tools as part of Montage

An important aspect of Montage is that, of its five core tools, two of them will be directly based on existing tools

for grammar engineering; these are the Grammar Matrix and the LKB/[incr tsdb()] tool combination, both developed as part of the LinGO (Linguistic Grammars Online) project.[8] Our use of such tools represents, we believe, an important convergence between the methods of computational linguistics and the methods of descriptive linguistics.

The Grammar Matrix (Bender et al., 2002) is designed to jump-start the process of implementing precision grammars by abstracting knowledge gained in grammar engineering activities done by the LinGO project into a form that can be easily reused by grammar engineers working on new languages. An early prototype of the Grammar Matrix is being used with promising results in the development of grammars for Norwegian (Hellan and Haugereid, 2003), Modern Greek (Kordoni and Neu, 2003), and Italian[9] (all funded by the EU Project 'DeepThought'[10]). Currently, the most elaborated portion of the Grammar Matrix is the syntax-semantics interface. This aspect of the core grammar assists grammar engineers in converging on consistent semantic representations for different languages.

The LKB grammar development environment (Linguistic Knowledge Builder; Copestake (1992, 2002)) , includes a parser and a generator as well as support for developing implemented formal (typed feature structure) grammars. [incr tsdb()] (Oepen, 2001) is a comprehensive environment for profiling and evaluating both grammars and parsing systems which is integrated with the LKB. The system design of Montage will allow the linguist to use the LKB/[incr tsdb()] tools directly, and, in addition, will provide for additional levels of functionality specifically designed to facilitate identification of candidate sentences for grammatical annotation.

While the tools developed by the LinGO project were designed with formal grammars in mind, they assume a model of grammar not dissimilar to that employed by the traditional field linguist, and, thus, can be directly applied to descriptive work. Specifically, both LinGO grammars and traditional descriptive grammars assume a rich category structure is operative in language and that grammatical description consists of generalizations over those categories. The main difference between descriptive grammars and the formal model of grammar employed by the LinGO tools is simply one of precision—in order to be machine readable, a restricted, well-defined set of categories must be rigidly employed for resources using the LinGO tools, while this requirement has not been essential for traditional grammatical description.

However, even though descriptive grammarians have not generally aimed for the level of precision required for computational applications, with the rise of the use of digital resources in all aspects of linguistics, efforts have begun to make descriptive materials precise in a way which would facilitate their being machine-readable.

The EMELD project's work on the GOLD ontology is a good example of research in this vein, since it is an attempt to codify traditional terminology into a well-defined controlled vocabulary of terms which can be used in all kinds of linguistic resources. In order to take full advantage of the accessibility provided by an ontology, we intend to support links to the ontology from both the descriptive and implemented grammars created with Montage. We expect that this work will place new demands on the GOLD ontology. Thus, while Montage has been made possible, to a large extent, by work on ontologies, we expect work developing the toolkit will also be valuable in refining and enhancing the ontologies themselves.

6. Conclusion

The goal of the Montage project is to make advances in electronic data management and computational linguistics accessible to field linguists working on the documentation of grammars of underdescribed languages. We envision two final products based on the resources of our toolkit. The first is the modern version of the traditional descriptive grammar. Without the inherent limitations of a paper-based format, these electronic grammars will allow easy access to the entire corpus of source examples, enhancing linguistic research. The second is a set of machine-readable resources codifying the grammatical analyses of the language. These resources will be valuable in linguistic hypothesis testing as well as practical applications such as machine translation or computer assisted language learning.

7. References

Bender, Emily M., Dan Flickinger, and Stephan Oepen, 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th Internation Conference on Computational Linguistics*. Taipei, Taiwan.

Copestake, Ann, 1992. The ACQUILEX LKB: Representation issues in the semi-automatic acquisition of large lexicons (ACQUILEX WP No. 36). In Antonio Sanfilippo (ed.), *The (other) Cambridge ACQUILEX papers*. University of Cambridge Computer Laboratory, Technical report No. 253.

Copestake, Ann, 2002. *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI Publications.

Farrar, Scott and Terry Langendoen, 2003. A linguistic ontology for the semantic web. *GLOT International*, 7:97–100.

Hellan, Lars and Petter Haugereid, 2003. Norsource – an exercise in the Matrix Grammar building design. In Emily M. Bender, Dan Flickinger, Frederik Fouvry, and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Engineering, ESSLLI 2003*.

Kordoni, Valia and Julia Neu, 2003. Deep grammar development for Modern Greek. In Emily M. Bender, Dan Flickinger, Frederik Fouvry, and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Engineering, ESSLLI 2003*.

Oepen, Stephan, 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany. In preparation.

[8] http://lingo.stanford.edu

[9] http://www.celi.it/english/hpsg_itgram.htm

[10] http://www.project-deepthought.net

Generic Morphological Analysis Shell

Akshar Bharati, Rajeev Sangal, Dipti M Sharma, Radhika Mamidi
Language Technologies Research Centre,
International Institute of Information Technology,
Gachibowli, Hyderabad – 500 019. India.
E-mail: {sangal,dipti,radhika_m}@iiit.net

Abstract

Morphological analyzers are mainly available for major languages like English, Spanish and French. This paper describes a generic shell which can be used to develop morphological analyzers for different languages particularly minority languages. The shell uses finite state transducers with feature structures to give the analysis of a given word. The most significant aspect of the shell is the integration of paradigms with augmented FSTs. The simple format in which a linguist may provide data in the form of dictionaries, paradigm and transition tables is the other important feature of the shell. The shell has been experimented on Hindi, Telugu, Tamil and Russian languages.

1. Introduction

A generic morphological analysis shell is being developed which will allow rapid development of morphological analyzers for different languages. The shell together with language data will take a word as input and return its morph analysis in terms of root and features. The current model is developed using Hindi, Telugu, Tamil and Russian sample data. The model will be tested on other languages before arriving at the final model.

2. Working of the Shell

The shell allows a series of transducers to be put together where the output of one phase becomes the input for the next phase. The advantage of multiple phases is that the language analysis can be done modularly. For example, the initial phase can break the input word into root and affixes. The subsequent phases can take these as input and do the analysis and return the appropriate feature structures. However, our formalism is different from KIMMO system as given by Koskenniemi (1983) and Antworth (1990) as described later.

The sample data from English, Hindi and Telugu shows such a break up. For these languages, the break up is done in two phases. The first phase is a root and suffix identifier and the second phase is a root and suffix analyzer (see figure 1).

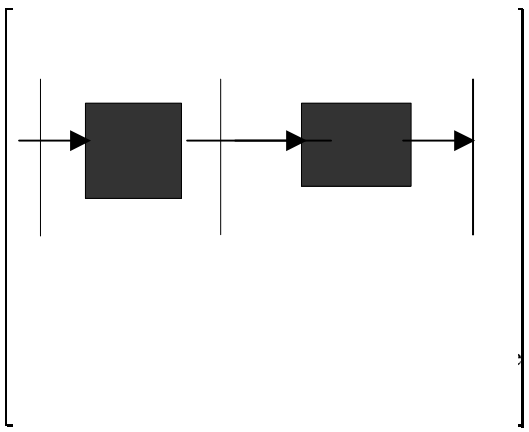


Figure 1: Transducer

For example, in case of English, given a word, say 'ate', as input, Phase 1 returns the root and suffixes with some obvious features as can be seen from the example given below. (Features are given in XML notation with 'fs').
For example: input string = 'ate'

Phase 1:
INPUT: ate
OUTPUT: eat<> +ed<>

This output is taken as the input for the next phase. The output of phase 2 is the final analysis of the given word.

Phase 2:
INPUT: eat<> +ed <>
OUTPUT: <fs root=eat category=verb tense=past>

This can be further illustrated with another input string 'am_going'

Phase 1:
INPUT: am_going
OUTPUT: be<fs gender=masculine|feminine number= singular person=first> go<> +ing<>

Phase 2:
INPUT: be<fs gender=masculine|feminine number= singular person=first> go<> +ing<>
OUTPUT: <fs root=go category=verb tense=present aspect=continuous gender=masculine|feminine number= singular person=first>

For implementing the above, we need at least two resources –

a. A generic shell which allows FSTs and other data to be defined.
b. Language specific data such as dictionaries, paradigm class and tables, and language specific FSTs.

Let us look at the formalism of the shell and the specifications for providing the data.

3. Formalism of the Shell

To introduce the augmented transition transducer, we begin by explaining the finite state transducers (FSTs). A FST has a set of states and directed arcs between them. The arcs are labeled by input symbols and possibly output symbols. One of the states is called the starting state. During a typical operation of the machine, it makes a transition from the current state to a new state by consuming a token from the given input string (of tokens) provided the input token matches the input symbol labeled on the arc. The output symbol on the arc is sent to output buffer.

The machine completes successfully if the machine is in an accepting state when the input string is fully consumed. Thus, the machine always begins its operation in the starting state and if on consumption of the input string, it ends in an accepting state, it is said to complete successfully. On a successful completion, the output buffer is actually sent to the output.

The formalism adopted here is augmented transition transducer with conditions, actions and feature structures. The finite state transducers are augmented with arcs labeled by the name of a specific FST, and feature structures with conditions and actions. Thus, they are like ATNs but also act as transducers.

The labels on the arcs may be a string, dictionary, paradigm or another FST.

The integration of paradigms with augmented FSTs is a unique feature of the shell. It is convenient to define paradigms for languages with relatively simple morphology such as Hindi, Bengali, etc. as shown in Bharati et. al. (1995; Chap. 3). One can give word forms of a prototypical root in a table (paradigm), and declare as similar other roots which behave like this one. It is left to the machine to compile add-delete strings from the table, to be used appropriately while processing actual word forms.

However, even the languages with simple morphology have some phenomena which are not handled by paradigms very satisfactorily. In such cases, usually, the number of entries increases in the table, and there are obvious patterns in the word forms which are not captured well by add-delete strings. Thus, the person giving the paradigm, although he knows or sees the patterns, cannot specify them using the paradigms. In the case of agglutinative languages like Telugu (Krishnamurti, 1985; Rao, 2002), the problem is even more acute, because a sequence of word forms or affixes is concatenated to yield the final word form. Each of the internal word forms can be described in paradigms. But there is no way to put them together using the paradigms framework.

In the work being reported here, the paradigms are integrated with augmented FSTs. This method allows paradigms to be used for describing a large part of the morphology, and the augmented FSTs to build on top by describing the hard cases including the concatenation of word forms. Thus, the simplicity of paradigms is combined with the power of AFSTs.

This integration is achieved by providing arcs which are labeled with paradigms. While traversing such arcs, the paradigms are looked up and all the search using add-delete strings performed in a transparent way.

Besides the above, the augmented FSTs can be put in a cascade where the output of one phase, feeds as input to the next phase. Such a feed is so designed that even the selected features can be transferred across. Such phases in the cascade can also be used to do chunking.

The data has to be provided following the given specifications for:

a. Dictionaries
b. Paradigm tables
c. Paradigm classes
d. FSTs (may be given as transition tables or drawn pictorially)

A. Dictionaries:

For an analysis of inflected or agglutinative word forms, a minimum of two dictionaries are required - root dictionary and affix dictionary. More dictionaries can be provided according to the requirement of the language.

A root dictionary contains a list of all the base forms of words with their feature structures, if any. The format is:

Root [TAB] Features

For example:

```
eat     <fs root=eat category=verb>
apple   <fs root=apple category=noun>
book    <fs root=book category=verb>
book    <fs root=book category=noun>
```

On the other hand, the affix dictionary contains all the prefixes, infixes and suffixes with their features. Generally these affixes express features like mood, aspect, tense, number, gender, person, negation, passivity, reflexivity, transitivity etc. Superfixes i.e. suprasegmentals like word stress, tone etc. which are essential for the description of words in languages like Russian or Ibibio may be included as well. The format of the affix dictionary is similar:

Affix [TAB] Features

For example,

```
+s      <fs tense = present>
+ed     <fs tense = past>
+en     <fs aspect = perfective>
+ing    <fs aspect = progressive>
```

In the dictionary, the features are given as attribute-value pairs separated by spaces within angular brackets '<>'. Optionality is indicated by '|'.

The embedded features should be nested within the main angular brackets as can be seen in the features of 'am', 'is' and 'are'.

```
am      <fs tense=present agr=<fs g=m|f n=sg p=1>>
are     <fs tense=present agr=<fs g=m|f n=sg p=2>>
is      <fs tense=present agr=<fs g=m|f|nt n=sg p=3>>
are     <fs tense=present agr=<fs g=m|f|nt n=pl
        p=1|2|3>>
```

where agr=agreement, g=gender, n=number, p=person, m=masculine, f=feminine, nt=neuter, sg=singular, pl=plural.

Zero suffixes may also be specified in the FSTs. They need not be mentioned in the dictionaries.

B. Paradigm tables:

When an arc labeled by a paradigm is called, then the related paradigm table and paradigm class are looked up.

The paradigm tables are associated with prototypical roots and are necessary to cover all the possible morphological processes dealing with morphophonemics when affixation takes place. Some of the morphological processes that can be handled by the shell are:

a. **Suppletion** (Replacement of the entire word form)
   Example: English
   ate – the past tense form of 'eat'
   is – the singular present tense form of 'be'

b. **Assimilation** (Change in sound due to the influence of neighbouring sounds)
   Example: English
   in + logical → illogical
   in + possible → impossible

c. **Apocope** (Deletion of final sound or syllable)
   Example: Telugu
   ceTTu + lu → ceTlu ("trees")

d. **Syncope** (Deletion of medial vowels or consonants)
   Example: Hindi
   asala + I → aslI ("real")

e. **Epenthesis** (Addition of phoneme/s or syllables when two morphemes combine)
   Example: Telugu
   mA + Uru → mAvUru ("my village /town")

f. **Vowel Harmony** (Influence of one vowel on another vowel of the preceding or following syllables)
   Example: Telugu
   maniSi + lu → manuSulu ("people")

These changes can be represented in the paradigm table as shown. The surface form followed by a TAB, followed by its analyzed components with features.
For example:

eats	eat<> +s<fs tense=present>
ate	eat<> +ed<fs tense=past >
mice	mouse<> +s<fs number=pl>
spies	spy<> +s<fs number=pl>

In our data, '+' is used with the suffixes. Therefore, plus sign '+' should not occur in input string.

C. Paradigm classes:

Along with the paradigm table, paradigm class has to be provided as well. A paradigm class contains the classes of words i.e. the prototypical root and all the roots that fall in its class including the given root. By the term 'root' we mean the base form or stem to which affixation takes place. The class is defined by enumerating roots in terms of their paradigm type i.e. those words which decline or conjugate in exactly the same way, fall into one type.

For example, the English verbs 'play' and 'look' have the following paradigm:

play	plays	played	played	playing
look	looks	looked	looked	looking

So they belong to the same class, whereas, 'push' since it differs in its present tense form i.e. it has '-es' and not '-s' falls in another class. Its paradigm is as follows:

push	pushes	pushed	pushed	pushing

As 'eat' is irregular verb, it belongs to a separate class of its own. Its paradigm is as follows:

eat	eats	ate	eaten	eating

Similarly, nouns, which decline in the same way, belong to one paradigm. For example, 'day' and 'boy' fall in one class as 'play', but 'spy' though ending in 'y' falls in another class since its plural is 'spies' and not 'spys'.

The data should be provided with the root, followed by a TAB, the POS tag, followed by TAB and then all the roots in the same class, separated by a comma as shown below.

Root [TAB]POS [TAB]root1, root2, root3,etc.

For example:

eat	V	eat
play	V	play, talk, walk, train
push	V	push, fish
play	N	play, boy, day
spy	N	spy, sky

D. FSTs:

The data for the AFSTs for affixation has to be provided in the form of transition tables. FST may be given pictorially or by transition tables which are its textual representation. The following is the format for the transition table.

The first line lists the initial states.
The second line lists all the states.
The third line lists the accepting states.
From the fourth line, the actual data is given using the following six fields separated by TABS on the same line:
SOURCE DESTINATION INPUT OUTPUT ACTION CONDITION

The last three fields should be given in angular brackets. A pictorial representation [FSTs] of all possible occurrences of concatenation of affixes for a given root will surely help in giving data in this format.
Example of a FST (as depicted in phase 2)

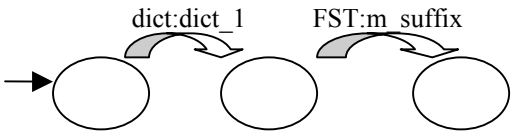
Phase 1:
INPUT: ate

OUTPUT: eat<> +ed<>
In this phase, the analysis is obtained as a result of the paradigm lookup.

Phase 2:
INPUT: eat<> +ed<>

In this phase the given sequence of strings traverse through the FSTs to give the final result. Given below are the pictorial and textual representations of the FST:m_main.

FST: m_main



S1
S1 S2 S3
S3
S1 S2 dict:dict_1 <> <> <>
S2 S3 FST: m_suffix <> <> <>

The first string 'eat' is consumed after checking for its validity in the dictionary. The rest of the sequence is passed through the FST named m_suffix to get the final output as:

OUTPUT: <fs root = eat category=verb tense = past>

The transition table for FST:m_suffix would be:

S1
S1 S2
S2
S1 S2 str:+@ <> <> <fs number=pl>
S1 S2 str:+s <fs tense=pres> <> <fs number=sg>
S1 S2 str:+ed <fs tense=past> <> <>
S1 S2 str:+en <fs aspect=perfective> <> <>
S1 S2 str:+ing <fs aspect=continuous> <> <>

Given below is an example from Telugu, with gloss, tested on the shell.
For example: input string = tin_i_vEs_EE_Du
"he completed eating"
Phase 1:
INPUT: tin_i_vEs_EE_Du
[GLOSS: tin = "eat", i=past participle, vEs= completive operator, EE=perfective aspect, Du=third person, singular, masculine marker]

OUTPUT: tinu<> i<> veyy<> EE<> Du <fs gender=m number=sg person=3>

Phase 2:
INPUT: tinu<> i<> veyy<> EE<> Du <fs gender=m number=sg person=3>

OUTPUT: <fs root=tinu category=verb tense=past aspect= perfective gender =masculine number=singular person= third>

4. Implementation

The implementation is especially designed to integrate handling of paradigms with the FSTs efficiently. Paradigm handling involves search, for which known algorithms are used (Bharati et. al., 1995), but integrated with the FSTs.

The augmented FSTs are non-deterministic, which are handled by the implementation appropriately. Feature structures are handled efficiently.

5. Conclusion

The purpose behind the design of such a powerful shell is to create a "free" tool which can be used by people across the world to build morphological and other analyzers for their languages. The shell is suitable for handling not just morphology but also doing chunking.

6. Acknowledgements

This work is supported by Outside Echo. We thank Roger Tucker of Outside Echo for his feedback. Ksenia Shalonova has worked out details of Russian and participated extensively in discussions.

7. References

Antworth, E. L. (1990). *PC_KIMMO: A Two-Level Processor for Morphological Analysis.* Summer Institute of Linguistics, Dallas, TX.

Bharati, Akshar, Vineet Chaitanya, Rajeev Sangal. (1995). *Natural Language Processing: A Paninian Perspective.* New Delhi: Prentice Hall. http://iiit.net/ltrc/nlpbook/index.html

Krishnamurti, Bh. and J. P. L. Gwynn. (1985). *A Grammar of Modern Telugu*. Delhi: OUP.

Koskenniemi, K. (1983). *Two-level morphology: A general computational model of word-form recognition and production*. Tech. rep. Publication No. 11, Department of General Linguistics, University of Helsinki.

Rao, G. U. (2002). *Compound Verb Construction in Telugu.* In V. Swarajya Lakshmi (Ed.), Case for Language Studies: Papers in honour of Prof. B. Lakshmi Bai (pp. 157--177). Hyderabad: Booklinks Corporation and Centre of Advanced Study in Linguistics, Osmania University.

Linguistic Resources and Tools for Automatic Speech Recognition in Basque

Bordel G.[1], Ezeiza A.[2], Lopez de Ipina K.[3], Méndez M.[3],

Peñagarikano M.[1],Rico T.[3], Tovar C.[3], Zulueta E [3].

University of the Basque Country

[1]Elektrizitate eta Elektronika Saila, Leioa. {german,mpenagar}@we.lc.ehu.es
[2]Ixa taldea. Sistemen Ingeniaritza eta Automatika Saila, Donostia. ispezraa@sp.ehu.es
[3] Sistemen Ingeniaritza eta Automatika Saila, Gasteiz. {isplopek,jtzmebej, iszripat,vcbtomoc,iepzugue}@we.lc.ehu.es

Abstract

The development of Automatic Speech Recognition systems requires of appropriated digital resources and tools. The shortage of digital resources for minority languages slows down the development of ASR systems. Furthermore, the development of the system in Basque represents even a bigger challenge, since it has a very uncommon morphology.

Nevertheless, recent advances have been achieved thanks to the Basque mass media, particularly the Basque Public Television (EITB) and the only daily newspaper in Basque (Egunkaria). These media have provided digital multimedia resources for the development of a digital library for both Basque and Spanish (Bordel, et al., 2004), and these rich resources have been employed to develop new tools for Speech Recognition.

Introduction

There is considerable current interest in development of digital resources and tools for Automatic Speech Recognition systems. Minority languages constitute a bigger exertion because of the lack of resources in general, and Basque is not an exception.

Recent works address this shortfall processing digital resources from Basque mass media. The interest is mutual, since mass media aim to employ Human Language Technology based applications to search and index multimedia information.

The purpose of the experiments reported in this paper is to develop appropriated resources for Automatic Speech Recognition in Basque. Both Basque and Spanish are official in the Basque Autonomous Community, and they are used in the Basque Public Radio and Television *EITB* (EITB) and in most of the mass media of the Basque Country (radios and newspapers).

Basque is a Pre-Indo-European language of unknown origin and it has about 1.000.000 speakers in the Basque Country. It presents a wide dialectal distribution, being six the main dialects. This dialectal variety entails phonetic, phonological, and morphological differences.

Moreover, since 1968 the Royal Academy of the Basque Language, *Euskaltzaindia* (Euskaltzaindia) has been involved in a standardisation process of Basque. At present, morphology, which is very rich in Basque, is completely standardised in the unified standard Basque, but the lexical standardization process is still going on.

The standard Basque, called "Batua", has nowadays a great importance in the Basque community, since the public institutions and most of the mass media use it. Furthermore, all the digital resources developed for this job are in this standard version of Basque, and thus, it has been used to develop all the new tools for Speech Recognition that we present in this report.

Besides, these new tools are classified in two sides. In the one hand, an automatic tool to generate appropriated Lexicons in Basque has been generated, in order to widely use the contents of the digital libraries.

In the other hand, the information extracted from the broadcast news videos and the newspaper texts has been used to develop a prototype of CSR system based on the HTK tool (Young, et al. 1997). The HTK tool-based system produced interesting results, but a more sophisticated approach was tried analysing the research developed for Japanese, a language that has a similar phonetic concerns to Basque. Thus, a prototype of simple tasks was developed based on Julius, a Multipurpose Large Vocabulary CSR Engine (Lee et al., 2001). This tool is based on word N-grams and context-dependent Hidden Markov Models, and the prototype produced significant results.

The following section describes the resources developed from the data provided by the aiding mass media. The third section presents the tools developed during this work, the fourth section describes the processing of the data, and finally, conclusions are summarised in the last section.

Multimedia Resources

In order to develop new tools for ASR systems, this project has demanded a previous work involving the development of richer digital resources. As it has been mentioned before, the main Basque media have collaborated in this development, providing videos from daily programs of broadcast news and newspaper texts.

Next follows a relation of the resources provided and created within the scope of this project:

- 6 hours of video in MPEG4 (WMV 9) format of "Gaur Egun" program, the daily program of broadcast news in Basque directly provided by the Basque Public Radio and Television (EITB).
- 6 hours of audio (WAV format) extracted from the video (MPEG4) files.
- 6 hours of audio transcription in XML format containing information about speaker changes, noises and music fragments, and each word's phonetic and orthographical transcription including word's lemma and Part-Of-Speech disambiguated tags.
- 1 year of scripts, in text format, of the "Gaur Egun" program.
- 1 year of local newspapers in Basque, Euskaldunon Egunkaria (Egunkaria), in text format.
- Lexicon extracted from the XML transcription files, including phonological, orthographical, and morphological information.

Automatic Speech Processing tools

The Automatic Speech Processing tools developed have been based on existing tools for Basque. The improvements included in this late part of continuous development are specially linked with the new resources adverted in the previous section. As it has been brought up in the introduction, two are the main lines of work: Lexicon development tools and Continuous Speech Recognition engines.

Lexicon development tools

Lexicon development has critical dependency with morphological features of a given language. Namely, Basque is an agglutinative language with a special morpho-syntactic structure inside the words (Alegria et al., 1996) that may lead to intractable vocabularies of words for a CSR when the size of task is large.

The lexicon extracting tool for Basque *AHOZATI* (Lopez de Ipina et al., 2002) tackles this problem using morphemes instead of words in order to define the system vocabulary. This approach has been evaluated over three textual samples analysing both the coverage and the Out of Vocabulary rate, using words and pseudo-morphemes obtained by the automatic morphological segmentation tool. Fig. 1 and Fig. 2 illustrate the analysis of Coverage and Out of Vocabulary rate over the textual sample from the broadcast news scripts. When pseudo-morphemes are used, the coverage in texts is better and complete coverage is easily achieved. OOV rate is higher in this sample.

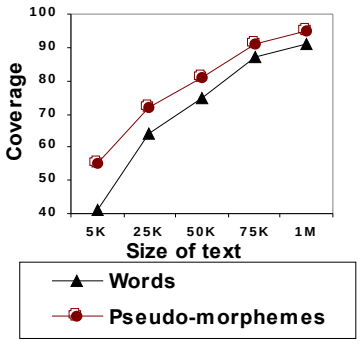


Fig. 1: Coverage for the textual sample.

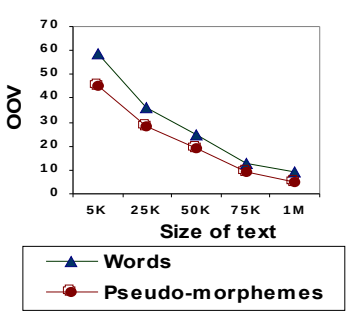


Fig. 2: OOV rate for the textual sample.

In addition, AHOZATI has been improved during this project extracting the lexicon from the XML transcription files mentioned in the *Multimedia Resources* section.

CSR systems

Hidden Markov Models are widely used in Continuous Speech Recognition systems, and toolkits like HTK (Young, et al. 1997) are well known and tested. Moreover, the prototypes we have developed yet have been principally adapted to the HTK standards. Thus the first use given to the recently obtained enriched lexicon was to be adapted to the HTK toolkit.

The HTK tool-based system produced interesting results, but due to its limitations both in licensing terms and language modeling, a more refined approach was tried utilising the open source tools developed for Japanese, a language that has a similar phonetic concerns to Basque.

Thus, a prototype of simple tasks was developed based on Julius, a Multipurpose Large Vocabulary CSR Engine (Lee et al., 2001). This tool is based on word N-grams and context-dependent Hidden Markov Models, and uses similar input files to HTK. The use of pseudomorphemes and N-grams fits in Basque much more than the word approach of HTK. This conclusion has appeared in our first experiments, although further evaluation has to be done.

Processing Methodology

Processing of the video data

The video data used in this work has been provided directly by the Basque Public Radio and Television. The format used to store the broadcast contents is MPEG4 (WMV 9), and the Basque Public Radio and Television has been very kind offering us all these resources.

Processing of the audio data

The audio data has been extracted out from the MPEG4 video files, using FFmpeg free software[1]. The audio files have been stored in WAV format (8 KHz, 16 KHz, linear, 16 bits).

When the audio data was ready, the XML label files were created manually, using the Transcriber free tool (Barras et al., 1998). The XML files include information of distinct speakers, noises, and paragraphs of the broadcast news. The files also contain phonetic and orthographic information of each of the words. Basque XML files include morphological information such as each word's lemma and Part-Of-Speech tag.

The Lexicon aforementioned has been extracted using this transcribed information. The Lexicon stores information of each different word that appears in the transcription.

Processing of the textual data

There are two independent types of textual resources: The text extracted from the newspaper Euskaldunon Egunkaria (Egunkaria), and the scripts of the "Gaur Egun" program.

All of the texts were processed to include morphologic information such as each word's lemma and Part-Of-Speech tag. Using all the information, a Lexicon for each language has been extracted taken into account the context of the word in order to eliminate the ambiguity. This Lexicon differs from the Lexicon extracted from the transcription files, and it is been developed to be used in testing and evaluating scenarios for Machine Learning techniques.

Concluding Remarks

This work deals with the development of appropriated resources and tools for an automatic index system of broadcast news in Basque. Since Basque is an agglutinative language, analysis of coverage and words OOV has been carried out in order to develop appropriated Lexicon. New resources were developed during this work. Subsequently, the Lexicon Extraction tool AHOZATI was improved with the new resources and finally two CSR prototypes were developed based on Hidden Markov Models. First, a HTK toolkit-based word prototype, and second, a Julius toolkit-based N-gram prototype.

Acknowledgments

We would like to thank all the people and entities that have collaborated in the development of this work, specially: EITB, Gara and Euskaldunon Egunkaria.

References

EITB Basque Public Radio and Television, *http://www.eitb.com/*

Euskaltzaindia, *http://www.euskaltzaindia.net/*

Peñagarikano M., Bordel G., Varona A., Lopez de Ipina: "Using non-word Lexical Units in Automatic Speech Understanding", Proceedings of IEEE, ICASSP99, Phoenix, Arizona.

[1] Available on-line at http://ffmpeg.sourceforge.net

Lopez de Ipiña K., Graña M., Ezeiza N., Hernández M., Zulueta E., Ezeiza A., Tovar C.: " Selection of Lexical Units for Continuous Speech Recognition of Basque", Progress in Pattern Recognition, pp 244-250. Speech and Image Analysis, Springer. Berlin. 2003.

Lopez de Ipina K., Ezeiza N., Bordel. N., Graña M.: "Automatic Morphological Segmentation for Speech Processing in Basque" IEEE TTS Workshop. Santa Monica USA. 2002.

Egunkaria, Euskaldunon Egunkaria, the only newspaper in Basque, which has been recently replaced by Berria, on-line at *http://www.berria.info/*

Barras C., Geoffrois E., Wu Z., and Liberman M.: "Transcriber: a Free Tool for Segmenting, Labelling and Transcribing Speech" First International Conference on Language Resources and Evaluation (LREC-1998).

Alegria I., Artola X., Sarasola K., Urkia M.: "Automatic morphological analysis of Basque", Literary & Linguistic Computing Vol,11, No, 4, 193-203, Oxford Univ Press, 1996.

Young S., J. Odell, D. Ollason, V. Valtchev, P. Woodland, *The HTK BOOK, HTK 2.1 Manual*, 1997

A. Lee, T. Kawahara and K. Shikano. "Julius --- an open source real-time large vocabulary recognition engine."In Proc. European Conference on Speech Communication and Technology (EUROSPEECH), pp. 1691--1694, 2001.

Bordel G., Ezeiza A. , Lopez de Ipina K., Méndez M. ,Peñagarikano M., Rico T., Tovar C., Zulueta E."Development of Resources for a Bilingual Automatic Index System of Broadcast News in Basque and Spanish", Fourth International Conference on Language Resources and Evaluation (LREC-2004).

Building Cat3LB: a Treebank for Catalan

Montserrat Civit, Núria Bufí and Ma. Pilar Valverde

CLiC, Centre de Llenguatge i Computació
Universitat de Barcelona
C/ Adolf Florensa s/n (Torre Florensa)
08028 Barcelona
{civit, nuria, pilar}@clic.fil.ub.es

Abstract

In this paper we present some specific issues related to the syntactic annotation of Cat3LB, a 100,000-word Catalan treebank (2,500 sentences). In the syntactic annotation we follow an incremental process with different levels of complexity: bracketing and labelling of constituents and functional tagging. Some automatic pre-processing steps have been done: morphological analysis, tagging and chunking. In this paper we will concentrate, however, on the syntactic annotation.

1. Introduction

Catalan is a 10-milion speaker language spoken in four different countries (Spain, France, Italy and Andorra), but it only has an official status in one of them (Andorra) which is the smallest one. One way to escape from the increasing pressure from the major languages and to avoid the demise of Catalan is to develop basic language resources. In this field, Catalan is not in a bad situation, since basic tools and resources have been developed so far (see section 2.). However, Catalan did not have one of the most important resources needed both to develop NLP applications and to acquire linguistic knowledge about how a language is used: a Treebank.

In this paper we present the development of a freely available treebank for Catalan: **Cat3LB**, built within the **3LB** project, whose goals are to build three treebanks: one for Catalan (**Cat3LB**), one for Spanish (**Cast3LB**) and finally another for Basque (**Eus3LB**). The **3LB** project is funded by the Spanish government[1].

Section 2. describes the previous processes and tools that have been used in the construction of the Treebank. Section 3. presents the main characteristics of the syntactic annotation carried out in the Cat3LB Treebank. Finally, section 4. comes to conclusions.

2. Previous Processes

CLiC-UB[2] and TALP-UPC[3] groups have developed so far a framework for the automatic processing of Catalan and Spanish, based in a pipeline structure shown in figure 1 (Atserias et al., 1998).

These tools include: a morphological analyser (MACO+), a morphosyntactic tagger (RELAX) and a chunker (TACAT). As the tagger is a probabilistic one, the **CLiC-TALP-CAT** corpus was developed in order to allow the tagger to learn the rules for the disambiguation. Then this corpus has been used to build the Treebank. The current CLiC-TALP-CAT corpus consists of 100,000 words coming from, on the one hand, the EFE news agency (25,000 words) and, on the other, from the ACN -Catalan News Agency- (75,000 words). All texts were originally written in Catalan.

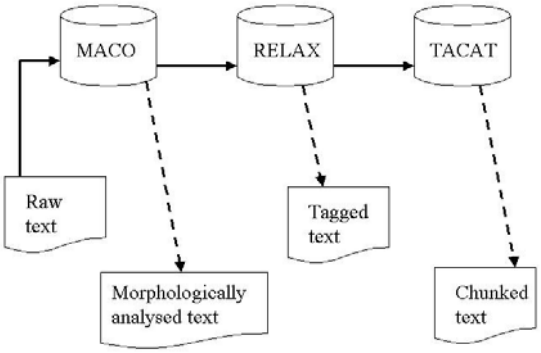


Figure 1: Pipeline of the NLP tools

MACO+ is a Morphological Analyser for Catalan which provides both lemma(s) and POS-tag(s) for each word. Its output has the following format:

$$word \quad lemma_1 - tag_1 \quad ... \quad lemma_n - tag_n$$

Tags codify 13 part-of-speech categories (noun, verb, adjective, adverb, pronoun, determiner, preposition, conjunction, interjection, dates, punctuation, numbers and abbreviations) as well as subtypes and morphological features, as it is proposed by Eagles (Monachini and Calzolari, 1996). The total amount of tags is 321 and they can be found, with a large explanation, at http://clic.fil.ub.es/doc/categorias_eagles_cat03.htm

The morphological analysis and lemmatisation of the sentence *El secretari general d'ERC ha participat aquest migdia en un acte davant la Delegació d'Hisenda de Tarragona per reclamar un finançament just per Catalunya*[4] is as follows:

El el DA0MS0 ell PP3MSA00

[1]PROFIT project FIT-1505000-2002-244. The development of tools and resources presented here has also been funded by XTRACT-2 (BFF2002-04226-C03-03).

[2]URL:http://clic.fil.ub.es/index_en.shtml

[3]URL:http://www.talp.upc.es/TALPAngles/index.html

[4]*The secretary general of ERC has taken part at noon in an act in front of the tax office in Tarragona in order to claim a fair funding for Catalonia.*

```
secretari secretari_general LI-NCMS000 secretari NCMS000
general secretari_general LF-NCMS000 general AQ0CS0 general NCCS000 general NCMS000
d' de SPS00
ERC ERC NP00000
ha ha I Ha NCMS000 haver VAIP3S0 hectàrea NCMN000
participat participar VMP00SM participat AQ0MSP
aquest aquest DD0MS0 aquest PD0MS000
migdia migdia NCMS000
en en DA0MS0 en PP3CN000 en SPS00
un 1 Z un DI0MS0 un DN0MS0 un PI0MS000 un PN0MS000
acte acte NCMS000
davant davant NCMS000 davant RG davant SPS00
la el DA0FS0 ell PP3FSA00 la I la NCMS000 La NCMS000
Delegació_d'_Hisenda_de_Tarragona Delegació_d'_Hisenda_de_Tarragona NP00000
per per NCMS000 per SPS00
reclamar reclamar VMN0000
un 1 Z un DI0MS0 un DN0MS0 un PI0MS000 un PN0MS000
finançament finançament NCMS000
just just AQ0MS0 just NCMS000 just RG
per per NCMS000 per SPS00
Catalunya Catalunya NP00000
```

After the morphological analysis, a constraint-based probabilistic tagger (RELAX) selects the most correct lemma-tag pair according to the near context (it mainly uses the preceding and the following word). Constraints have been automatically infered from the CLiC-TALP-CAT corpus, whose desambiguation was manually validated, and, in order to improve the accuracy, some handwritten rules have been added, mainly refering to the lemma ambiguity as well as to the subcategory one. Once the sentence has been disambiguated, the tagged text is as follows:

```
El el DA0MS0
secretari_general secretari_general NCMS000
d' de SPS00
ERC ERC NP00000
ha haver VAIP3S0
participat participar VMP00SM
aquest aquest DD0MS0
migdia migdia NCMS000
en en SPS00
un un DI0MS0
acte acte NCMS000
davant davant SPS00
la el DA0FS0
Delegació_d'_Hisenda_de_Tarragona Delegació_d'_Hisenda_de_Tarragona NP00000
per per SPS00
reclamar reclamar VMN0000
un un DI0MS0
finançament finançament NCMS000
just just AQ0MS0
per per SPS00
Catalunya Catalunya NP00000
```

Finally, the chunking with TACAT and a context free grammar for Catalan of about 1920 handwritten rules provides us with the following analysis:

```
S_[
sn_[ espec-ms_[ El_da0ms0 ]
     grup-nom-ms_[ secretari_general_ncms000
                   sp-de_[ prep_[ d'_sps00 ]
                           sn_[ grup-nom-fp_[ ERC_np00000 ] ] ] ] ]
grup-verb_[ ha_vaip3s0 participat_vmp00sm ]
sn_[ espec-ms_[ aquest_dd0ms0 ]
     grup-nom-ms_[ migdia_ncms000 ] ]
     sp_[ prep_[ en_sps00 ]
          sn_[ espec-ms_[ un_di0ms0 ] grup-nom-ms_[ acte_ncms000 ] ] ]
     sp_[ prep_[ davant_sps00 ]
     sn_[ espec-fs_[ la_da0fs0 ]
          grup-nom-fs_[ Delegació_d'_Hisenda_de_Tarragona_NP00000 ] ] ]
     sp_[ prep_[ per_sps00 ] S-NF-C_[ infinitiu_[ reclamar_vmn0000 ] ] ]
     sn_[ espec-ms_[ un_di0ms0 ] grup-nom-ms_[ finançament_ncms000
                                    s-a-ms_[ just_aq0ms0 ] ] ]
     sp_[ prep_[ per_sps00 ] sn_[ grup-nom-fp_[ Catalunya_np00000 ] ] ] ]
```

The construction of the treebank itself has consisted of the manual embedding of the chunks as well as of the functional tagging.

3. Syntactic Annotation

The task of manually building a treebank requires a tool for facilitating annotators' work. After looking for different freely available interfaces, we decided to use the **AGTK** toolkit set up by the Pennsylvania University (Cotton and Bird, 2000). The main advantage was that it could easily accept our chunker output as well as our large tagset. Figure 2 shows this interface. The main utilities of such an interface are that it allows to move, remove, adjoin and add nodes and tags; it also lets us split and merge sentences and words. Crossing branches is not allowed, so discontinuous constituents are given special tags, as we will comment further on.

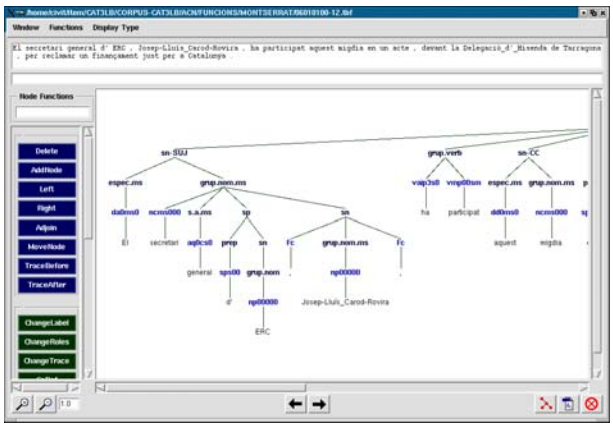


Figure 2: AGTK Interface

3.1. Annotation process

In a first step, 25.000 words were syntactically annotated in parallel (constituent annotation) by two linguists. This first annotation was then used, on the one hand, to refine the annotation criteria and, on the other, to enlarge the annotation guidelines previously established[5]. The comparison between the two annotations gave the results shown in table 1, in which **LP** stands for *labelled precision*; **BP** for *bracketed precision* and **CB** for *consistent bracketing*[6].

LP	0.876478
BP	0.90953004
CB	0.943214
same-lenght Sentences	
LP	0.9198125
BP	0.93964505
CB	0.96512s0

Table 1: Annotators' agreement

One of the main sources of disagreement among the annotators was whether to consider as a single word certain complex structures such as *des que* ('since'), *donar lloc a* ('give rise to') and so on. Annotators adopted different criteria when labelling and bracketing these units. This affected the lenght of the sentences: if such expressions were taken as multi-words, there were fewer terminal elements (words) in the sentence than if they were taken separately. Since our measures take into account the starting and finishing points of each constituent in the sentence, the fact that the length of the sentence varied implied a substantial decrease of the agreement measures. This issue has been accurately analysed and very strict criteria to deal with multiwords were established in the guidelines. However, our aim has been to evaluate also the agreement figures obtained only for those sentences whose lengths coincided. After the correction of these discrepancies, results improved significantly, even though a full agreement seems impossible.

[5](Valverde et al., 2004) is the last version of the guidelines for the constituent annotation.

[6]As they are used in Parseval.

The remaining corpus has been then annotated by only one linguist.

Once the constituent annotation was finished, we started the functional tagging. The process was the same: annotation in parallel of 10,000 words by two linguists; comparision of the results[7]; discussion of the differences; enlargement of the guidelines[8]; annotation by one linguist on the remaining sentences.

Up to now, the full corpus has been annotated at the constituent level, and half

of it at the functional one. At the first level, annotators spent one hour to annotate ten sentences; while at the second one, they needed the same time to annotate 25 sentences. The average number of words per sentence is 39.

3.2. Constituent annotation

General principles have been defined to be the main guidelines in the constituent annotation process. Firstly, we only mark explicit elements, although we add nodes for elliptical subjects, as in the initial purpose of the project it was settled that anaphoric and coreferential information would be annotated. As for elliptical finite verbs, instead of adding a new node, we mark sentence nodes with a suffix *****; that usually happens in coordinated structures, for which the second one is verbless.

Secondly, we follow the constituency annotation scheme instead of the dependency framework, since this is the framework that best matches Catalan morphosyntactic features.

Thirdly, we do not alter the surface word order so as to maintain pragmatic information, even though it implies to face the problem of discontinuous elements. We do not consider *verbal phrase* as a node containing the verb and its complements, but as a node containing only simple verbal forms (*considera*, '(he/she) considers') and complex ones (*ha participat*, '(he/she) has participated'; *ha estat considerat*, '(he/she/it) has been considered').

Cases of discontinuity have been dealt with in two different ways: some of them at the constituent level, and the others at the functional one. Discontinuity dealt with in the first level[9] is mostly related to the noun phrase and involves a noun complement which is separated from the head by a (verbal) complement, like in the sentence *Altae és l'aposta de banc de gestió de fortunes de Caja_Madrid en el qual l'entitat pretén guanyar-se una quota de mercat*[10] in which the relative clause *en el qual ...* depends on the noun *banc* but is separated from it by the complement *de Caja_Madrid*. In this case, we add an index **.1** to both elements involved in the discontinuity, so that the resulting analysis is as shown in figure 3.

As for concrete aspects of the annotation, we distinguish among different types of clauses: finite and non-finite

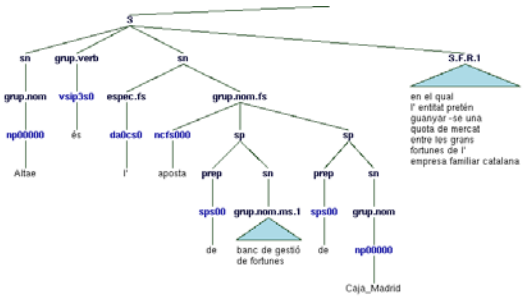


Figure 3: Constituent discontinuity-1

ones, on the one hand, and, on the other, completive, relative and adverbial ones. Adverbial clauses, moreover, are splitted into two groups: those considered as having a function as verbal complement (namely those meaning time, place, cause, purpose or manner) and those considered to be adjuncts of the verb (conditional, concessive, comparative and consecutive ones).

As it occurs in the PennTreeBank, we do use an equivalent to the PRN node for nodes that, generally speaking, do not belong to the sentence but to the discourse. In our case the tag is INC and mainly appears with reported speech.

We pay special attention to the treatment of coordination structures: we consider coordinated elements to be equivalent in the syntactic structure, so they are represented as siblings, which means that there is no head in such constructions. Shared complements are another issue related to coordination (i.e.: complements shared by two or more verbs); in these cases our solution is to adjoin the complement to the coordinated node.

Finally, we should point out that since punctuation marks are considered one among other elements of the sentence and receive a pos-tag, criteria to deal with them have to be clearly established, especially because they play a crucial role in the delimitation of the constituents (bracketing). We distinguish two main kinds of punctuation marks; those having a parenthetic role and those acting as a delimiter. The former appear as the first and the last element of the constituent, while the latter is the first element.

3.3. Functional tagging

Only daughter nodes of sentences and clauses are given a functional tag (i.e. we do not deal with noun complements). Given tags are subject, direct or indirect object, prepositional complement, adverbial complement, etc. and only surface functions are marked; that means that, for instance, we do not tag the subject of the infinitives depending on a control verb.

We have established a set of 15 basic tags, in order to cover all syntactic functions, and then, given specific marks (tag suffixes) to some of them in order to annotate specific cases of these functions. All in all, the total amount of tags at this level is 55.

The remaining case of discontinuity is dealt with in the functional tagging. An example of such a case is *només en queda un*[11] in which "en" and "un" are the two elements of the subject. In these cases, we add the suffix **.d** to the

[7] We do not have exact figures of the annotators' agreement, but they reached about 90% in the first sentences and more than 95% in the lastest ones.

[8] (Civit et al., 2004) is the guidelines for the functional tagging.

[9] See section 3.3. for discontinuity at the functional level.

[10] 'Altae is the bet for fortune managing of Caja_Madrid in which the company wants to gain market share'.

[11] Literally: 'only PRO there is one'; 'there is only one left'.

functional tag. This suffix must appear at least twice, one in each of the members of the splitted constituent.

Another case of discontinuity appears in relative or interrogative clauses, in which the relative (or interrogative) pronoun of a (non-)finite clause raises to the first position of the sentence, like in the sentence in figure 4: *dels pagesos que hi vulguin anar*[12], in which the selected locative complement (*hi*) belongs to the non-finite clause *anar* but appears before the main verb. For these cases, the functional tag has a suffix **.F** or **.NF** (depending on the type of the clause -finite or non-finite-) and the whole tag must be read as follows: *complement of the first finite or non-finite clause to the right*.

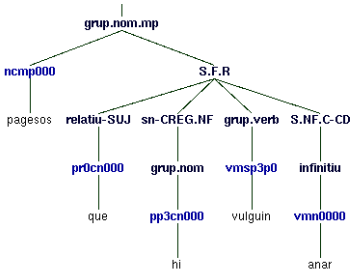


Figure 4: Constituent discontinuity-2

Sometimes a functional element appears two times (is repeated in the sentence). It usually happens with direct an indirect objects, when the phrase goes before the verb and it has to be repeated by a clitic, like in *El rànquing l' encapçala la final de la Champions_League*[13], in which the direct object appears twice at the beginning of the sentence (see figure 5), and the tag for the direct object has the suffix **.r**.

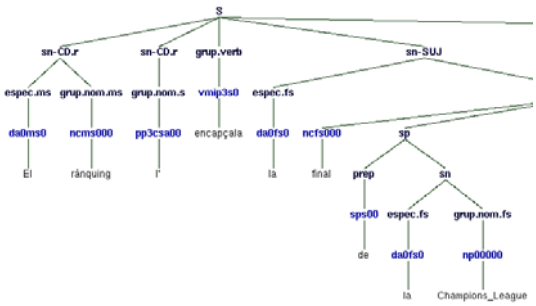


Figure 5: Double functions

One of the most controversial points related to functional tagging has been the distinction between prepositional complements selected or not by the verb. Linguistic criteria are not unanimous, especially those concerning the obligatoriness of the complement. It usually happens that locative complements are mandatory. Bearing in mind the state of the art about this point, we decided to give the adverbial tag (**-CC**) to those elements being optional, whilst the function tag **-CREG**, standing for 'selected PP' is used for the mandatory complements, no matter whether they are locative or not.

4. Conclusions and further work

We have presented different tools and resources developed so far for Catalan and paid special attention to the Catalan Treebank (**Cat3LB**): a morphological analyser (MACO+), a tagger (RELAX), a chunker (TACAT), a context free grammar and a manually validated corpus with morphosyntactic annotation (CLiC-TALP-CAT). All this resources are free for research purposes[14] and are intended to encourage linguistic and computational research on Catalan.

As further work and within the 3LB project, a subset of 10,000 words of the Treebank is being semantically annotated with Catalan-EuroWordNet (Benítez et al., 1998) and will also be freely available.

5. References

Atserias, J., J. Carmona, I. Castellón, M. Civit S. Cervell, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo, 1998. Morphosyntactic analysis and parsing of unrestricted spanish text. In *Proceedings of the First Conference on Language Resources and Avaluation. LREC'98*. Granada.

Benítez, L., G. Escudero, M. López, G. Rigau, and M. Tailé, 1998. Methods and tools for building the catalan wordnet. In *Proceedings of the ELRA Workshop on Language Resources for European Minority Languages, at the First Conference on Language Resources and Avaluation. LREC'98*. Granada.

Civit, M., N. Bufí, and M.P. Valverde, 2004. Guia per a l'anotació de les funcions sintàctiques de cat3lb: un corpus del català amb anotació sintàctica, semàntica i pragmàtica. Technical report, CLiC. Available: http://www.clic.fil.ub.es/personal/civit/publicacions.html.

Cotton, S. and S. Bird, 2000. An integrated framework for treebanks and multilayer annotations. In *Proceedings of the Second International Conference on Language and Evaluation LREC-2000*. Athens, Greece.

Monachini, M. and N. Calzolari, 1996. Synopsis and comparison of morphosyntactic phenomena encoded in lexicons and corpora. a common proposal and applications to european languages. Technical report, EAGLES. Available: http://www.ilc.pi.cnr.it/EAGLES96/browse.html.

Valverde, M.P., M. Civit, and N. Bufí, 2004. Guia per a l'anotació sintàctica de cat3lb: un corpus del català amb anotació sintàctica, semàntica i pragmàtica. Technical report, CLiC. Available: http://www.clic.fil.ub.es/personal/civit/publicacions.html.

[12] Lit: 'from farmers who there want to go'; translation: 'from farmers who want to go there'

[13] Cat:
El rànquing-CD l'-CD encapçala la final de la Champions_League
Lit.: 'The ranking-CD PRO-CD heads the final-SUBJ of the Champions League'
translation: 'the final of the Champions League heads the ranking'

[14] The mail contact is civit@clic.fil.ub.es

From Legacy Lexicon to Archivable Resource

Mike Maxwell

Linguistic Data Consortium
3600 Market St, Suite 810
Philadelphia, PA 19104 USA
maxwell@ldc.upenn.edu

Abstract

A common format for lexicons produced in field linguistics projects uses a markup code before each field. The end of each field is implicit, being represented by the markup code for the next field. This markup format, commonly called "Standard Format Code(s)" (SFM), is used in one of the most common lexicography tools used by field linguists, Shoebox. While this plain text format satisfies many of the desiderata for archival storage of language materials (as outlined in Bird and Simons 2003), there are usually problems with such lexicons as they are produced in practice which detract from their value.

In particular, SFM-coded lexicons commonly suffer from inconsistencies in the markup codes, especially in terms of the adherence of the fields to a hierarchical order (including omission of fields required by the presence of other fields). It is also common for the contents of certain fields to be limited to a fixed set of items, but for the lexicographer to have been inconsistent in the spelling of some of those items. Finally, spell checking (and correction) needs to be carried out in various languages, including both the glossing language(s) and the target language (where possible).

This paper outlines some tools for correcting these problems in SFM-coded lexicons.

Introduction: The Problem

One of the most important results of a typical field linguistic program is a bilingual dictionary. Most dictionaries are prepared in electronic format, often in the flat text format. Other formats can generally be converted into a plain text format.

At present, the most common flat file format is that produced by the SIL program Shoebox. This format utilizes a markup code at the beginning of each field; often this code begins with a backslash, e.g. "\w " for a headword. These tags are therefore known as "backslash markers", or more formally as "Standard Format Markers" (SFMs).

The end of a field is only marked implicitly, by the SFM of the following field. (Fields may occupy more than one line; normally, newlines within fields have no meaning.)

The beginning of a record is marked by the presence of a designated SFM (often either that of the headword field or an arbitrary record number, so that the designated SFM performs a dual function as field marker and record marker). The end of a record is marked by the beginning of the next record. (Often there is a blank line separating records, but this is neither sufficient nor necessary.)

While plain text format satisfies many of the desiderata for archival storage of language materials (as outlined in Bird and Simons 2003), there are certain typical problems with such SFM-coded lexicons as they are produced in practice which detract from their value.

One such problem has to do with the fact that dictionaries are actually structured objects, with logical constraints on the structure of fields within a record (lexical entry), the relationships between lexical entries, and on the contents of the fields themselves. While the structure can be represented using appropriate markup, in practice field linguists' lexicons violate the constraints, both at the level of the markup and at the level of the contents of the fields.

LinguaLinks (another SIL program) has a built-in model of lexical entries which enables it to impose well-formedness constraints at data entry time. However, LinguaLinks does not enjoy the large market share among field linguists that Shoebox does. While it is possible to impose some constraints on a Shoebox dictionary at data entry time, it is possible to do more validity checking in batch mode, provided there is a model of the lexicon. Such a model implies making explicit the semantics of the fields, a semantics which is implicit (albeit sometimes imperfectly so) in the user's mind when he (or someone else) designed the database.[1]

The purpose of this paper is to demonstrate the use of automatic validity checkers which can be applied off-line to an SFM-coded lexicon, marking up the database for errors in batch mode; the errors can then be searched for and corrected on-line. These checkers (or their predecessors) have proven useful in practice on a variety of text-based lexicons. The checks performed include:

- Verifying the markup codes, including their relative ordering and hierarchy (as specified by a model);
- Listing the parts of speech and other restricted fields, with occurrence counts (useful for finding erroneous field content);
- Doing spell for data in languages for which a spell checker is available, and character n-gram checking for languages for which no spell checker is available.

[1] There are in fact several well thought-out models of lexical databases which could be applied to the problem. Generally these models are hierarchical (e.g. senses within lexical entries), but they usually allow for cross-references as well (e.g. synonymy relations, major-minor lexical entries). This is well-suited to an XML structure. Unfortunately, while Shoebox has an XML export capability, it does not create a DTD or Schema, and there are some problems with its XML export (see e.g. http://www-nlp.stanford.edu/kirrkirr/dictionaries/).

In addition, I demonstrate a way to export the lexicon to Microsoft Word format, automatically marking the fields for their language so that the multi-lingual spell correction tools of Word can be applied.

There are other consistency checks which could also be performed. Shoebox has the built-in capability of checking that for every cross-reference, the target of that cross-reference exists. However, Chris Manning (p.c.) has suggested that one should also check for bidirectional references (e.g. synonyms), and this checking capability is not built into Shoebox. This sort of check may be added to the suite of tools described here in the future. Another useful check would be that sense numbers begin with 1 and are sequential.

The validity checking tools will be made available at a public website.

Verifying Markup Codes

Version 5 of Shoebox[2] provides a number of checks that can help ensure consistency. Hence, while it is not necessary to use Shoebox to maintain an SFM-coded dictionary (or other database), Shoebox is a useful tool in the verification process.

Most of the consistency checks in Shoebox are set up using Shoebox's "Database Type Properties" dialog box. For example, Shoebox can be told which fields can be empty, and it will check for fields which should be filled, prompting the user to fill in the missing data. However, while Shoebox can be told which field should follow a given field, it only uses this information when the user adds a new field[3]; it does not check for missing fields which should follow a given field in existing data.

Hence, the first consistency check described here ensures that all required fields are present. It would be helpful if the information concerning the fields could be extracted from the dictionary's 'type' file.[4]

An example of the information in one record of the .typ file appears here:

```
\+mkr d
\nam Definition (English)
\lng English
\MustHaveData
\mkrOverThis w
\mkrFollowingThis dfr
\-mkr
```

The field labeled \mkrOverThis defines the parent SFM of the given SFM: in this case, a \d field appears under a \w field. Unfortunately, this is not sufficient to describe the notion of an obligatory field. That is, the presence of a given field implies the presence of its hierarchical parent, and the presence of an immediately following field (if any). But there is no way to encode the necessity for a field which must appear, but which may not appear immediately after a given field. For example, if a record must have a definition field following a part of speech field, but a usage comment may optionally intervene, there is no way to encode this in the .typ file.

Accordingly, the consistency check for required fields must use its own representation of the dictionary structure. It therefore employs a standard regular expression notation to encode both the hierarchy and the obligatoriness of field structure within records, and the record structure within a dictionary file.[5] The following is an example expression defining the field structure of a dictionary file (the full notation is given in the program documentation):

```
id
(w
   ( (pos defn  (ex exEn exFr)*  (syn)?)
   | (num  pos  defn (ex exEn exFr)*  (syn)?)+
   )
)+
```

This is interpreted as follows. A dictionary file begins with a single \id record. Each following record is marked by a \w field, and may contain either of two alternatives: One alternative contains a part of speech (\pos), definition (\def), zero or more example sentences (\ex), each of which must have both an English (\exEn) and a French (\exFr) translation), and an optional cross-reference to a synonym (\syn; the optionality is indicated by the question mark). The other alternative consists of a one or more senses (represented implicitly), each of which contains a sense number (\num), followed by the same contents as the first alternative.[6]

Notice that the topmost structure is defined at the level of a dictionary file, not the entire dictionary. For many dictionaries, no such distinction is relevant: the entire dictionary is contained within a single file. It is not

[2] All remaining references will be to version 5 of Shoebox. More recently a similar tool called 'Toolbox' has been released (see http://www.sil.org/computing/catalog/show_software.asp?id=79) I have not tested the techniques in this paper under Toolbox, however Toolbox claims to be upwards compatible from Shoebox, so the procedures should work. Toolbox also includes a verification mode for glossed interlinear text, a feature of earlier versions of Shoebox which was omitted from version 5.

[3] In fact, a required field is only added when one hits the Enter key after adding the parent field of the required field. For example, adding an example sentence field will not add a field for the translation of that example sentence until the user hits the enter key at the end of the example sentence. Users may not in fact hit the Enter key when adding fields, so missing fields can arise even after the hierarchy of fields have been established.

[4] The name and location of the type file is given in the Database Types dialog box (Projects menu), and is created by Shoebox from the information in the previously referred-to Database Type Properties dialog box. The latter should therefore be checked for accuracy. Since Shoebox builds the information in that dialog from the database itself, it may contain obsolete information (e.g. SFMs which were used in earlier stages of the work). An undocumented feature is that only those SFMs which are actually used in the dictionary appear in bold in the Database Type Properties dialog. In most cases, any non-bold markers should therefore be removed.

[5] Allowing alternative record structures within the lexicon allows for different kinds of entries, such as minor entries. It also allows for various bookkeeping records that Shoebox includes, primarily at the top of the file.

[6] There is obvious redundancy in this description, which could be eliminated by use of something like the Backus Naur Form. For the sake of readability, I have not employed such a notation.

uncommon, however, for larger dictionaries to be maintained in separate files. For purposes of field checking, however, it should be sufficient to process each such file separately, since records should not cross file boundaries.

The operation of the field checker is as follows: it first reads in the regular expression defining the lexicon structure. It then reads a lexicon file in. Following the SFM notation, records are assumed to be everything from the record-marking SFM in one record to the next record-marking SFM, or to the end of the file (where a record-marking SFM is any top-level SFM in the regular expression). Ambiguity is unlikely here, but the parser uses an anti-greedy algorithm: the first SFM which could begin a new record is assumed to do so. All fields encountered before the next record-marking SFM are assigned to the current record.

Within a single record, the checker then attempts to assign the field markers actually found to the expected field structure. In case of error, a fall-back algorithm is used which allows for the possibility of an inappropriately missing field. For instance, suppose the parser encounters the following structure:

```
\ex    Yax bo'on ta sna Antonio.
\exEn I'm going to Antonio's house.
\ex    Ban yax ba'at?
\exEn Where are you going?
\exFr  Ou allez-vous?
```

Given the field definition above, there is a missing \exFr field after the first \exEn field. The parser encounters the second \ex field when it is expecting to find a \exFr field. It assigns the existing \enEn field under the current \ex field, hypothesizes a missing \exFr sub-field, and then begins with the second found \ex field. By way of an error message, it prints out an error message in the hypothesized \exFr field:

```
\ex    Yax bo'on ta sna Antonio.
\exEn I'm going to Antonio's house.|
\exFr  ***Missing field inserted***
\ex    Ban yax ba'at?
\exEn Where are you going?
\exFr  Ou allez-vous?
```

Later, the user can search for the error strings (by default these are flanked by '***') and make the appropriate repairs.

In general, when the parser encounters an unexpected field, it assumes that a single field is missing, and attempts to repair the error by inserting the expected field, then resuming the parse with the next actual field. The reasoning here is that fields are more often missing than inserted or put in the wrong order.

However, not all parsing errors can be repaired in this way. If an unexpected field is encountered which cannot be repaired by inserting a single missing field before it, then the unexpected field is labeled with an error message, and the parser attempts to resume with the next field marker, ignoring the presumably erroneous one. Consider the following record, which is ill-formed in the light of the earlier definition:

```
\w    yax
\pos  AUX-V
\pos  Adj
\defn green
```

Since within a record only one \pos field is expected (in the absence of a \num field indicating multiple senses), the parser labels the second \pos field as erroneous, and attempts to resume parsing with the \defn field:

```
\w    yax
\pos  AUX-V
\pos  Adj  ***Erroneous field***
\defn green
```

If neither repair—insertion of a single field, or over-looking a single field—succeeds, then the parser issues a general error message "***Unable to parse record structure***", and resumes parsing with the next record.

Obviously this simple-minded error correction algorithm can go astray, but it flags many errors correctly, and when it cannot determine the cause of an error, it will at least tell the user that there is a problem in the record structure.

An alternative to using a special purpose parsing algorithm would be to export the dictionary as an XML file from Shoebox, and to use existing XML parsing tools. However, while Shoebox can export an XML file, it cannot import one. This approach would therefore require a separate XML lexicon viewer, with many of the capabilities of Shoebox built in; the user would have to locate an error in the XML viewer, then search in Shoebox for the same record in order to repair the error. By instead parsing the SMF-coded file directly and writing the error messages into the SFM file, the errors can be displayed directly in Shoebox.

Occurrence Counts

Shoebox can restrict the contents of designated fields to a certain set of elements, termed a "Range Set." This is useful for closed class items, such as parts of speech. The list of allowable elements can either be built by hand, or Shoebox will build it automatically from the actual elements found in the data. In my experience, if field linguists employ range sets at all, the latter is the way the sets are built—which means that any erroneous items in the data are automatically added to the range set.

A savvy user can examine the range set and remove any spurious items, then run a consistency check to repair any fields which violate the edited range set. But in fact, it often devolves upon a consultant to perform this check (if not to perform the repairs). While obvious errors are easy to spot (the use of both "Noun" and "noun", say), the consultant may not be familiar enough with the grammar of the language to notice other erroneous items in the range set. For this reason, it is useful to count the number of times particular elements in a given field appear, on the principle that what is rare is often an error.

There are many ways this can be done; I use a simple program (coded in Python) which counts all the strings appearing between a particular pair of regular expressions. For counting parts of speech, for example, the search expression has "^\\pos " (a "\pos" at the beginning of line)

to the left, and "$" (end of line) to the right. The resulting list can be perused for low-frequency items.

Spell Correction

Spell checking can easily be done for most major languages by extracting the text from fields which are in the desired language, and running the extracted text through an off-line spell checker (such as *aspell* or *ispell*[7]).

One problem with this approach is that SFM-coded fields may not be contained on a single line. This is particularly true of example sentences (or their translations into the glossing language(s)). It is therefore not sufficient to *grep* out the lines containing the desired SFM codes, without first normalizing the file(s) so that each field occupies a single line. Again, this can be done in a variety of ways; I use a simple Python program to combine all the fields of a given record onto a single line, then break the record up into fields again at SFMs. (I also tokenize the result into words, and sort them uniquely so that each word need only be checked once; abbreviations and the SFMs themselves can also be filtered out at this stage.)

Another detail that could cause problems is the encoding issue. Spell checkers assume a particular encoding, and if the Shoebox dictionary uses a different encoding, it would be necessary to run the text through an encoding converter (such as *iconv*[8]) prior to spell checking.

However, the biggest issue for spell *checking* of a multilingual dictionary is that it is cumbersome to do spell *correction*. That is, while *aspell* supports spelling correction of a monolingual file, it is not easy to merge the corrected result back into the SFM-encoded dictionary, even if one does not tokenize the extracted fields. Nor would it be straightforward to run *aspell* directly on the SFM-encoded files, precisely because they are multilingual, and there is no way to tell *aspell* what language a given field is in.

If there are only a small number of spelling errors, this is perhaps not an issue. One can extract the fields, run them through a spell checker to produce a list of misspelled words, then use Shoebox to search for each of the misspellings in situ.

But a dictionary I was recently working with prompted me to find another solution: the glosses were in both English and French, and the French glosses had been entered without accents or cedillas. Spell correction was therefore a massive exercise, involving not only correction of typos, but entering numerous accented characters.

The better solution involved exporting the SFM dictionary to Microsoft Word, running a program in Word to define the language for each field, and using Word's built-in French and English spell correctors on their respective fields. The French spell corrector made it trivial to add the accented characters. (Of course Word could not automatically correct words where two forms existed which differed only by the presence of accents: *a* 'has' and *à* 'to', for instance.) The file was then exported back into Shoebox.

Note that this process uses Word only as a temporary way of modifying the dictionary. It is not intended that any sort of editing, apart from spell correction, be performed in Word, thus avoiding the problems inherent in doing lexicography in a word processor (Bird and Simons 2003).

In more detail, the SFM language marking program is written in Word's Visual Basic programming language, and functions in effect as a Word macro. The user imports an SFM-coded file into Word, then launches the program from within Word.

The SFM language marking program parses the information on fields and the language that they are encoded in from the Dictionary Type file (see footnote 4), making certain assumptions. For example, Word has separate spelling dictionaries for several dialects of French; if the user specifies "French" in the type file, the import program assumes this means what Word calls "French (France)".[9] The SFM language marking program then automatically assigns the contents of each field in the SFM-coded file to the appropriate language. If a field uses the "Default" language, the program marks the field as not to be spell-checked. (The SFMs themselves are also marked not to be spell-checked.)

Once the program has assigned the field contents to the appropriate languages, the user can use Word's spell checking/ correction features to correct the spelling. When finished, the user saves the file as text, allowing it to be imported back into Shoebox.

Finally, not all languages of interest have spell checkers or correctors. In particular, it is unlikely that the target language of a minority language dictionary will have any spell checking facilities (and building an *aspell* dictionary from the contents of a bilingual dictionary is obviously not an option, since it is the bilingual dictionary itself that is to be checked!). However, what can be done is to extract the relevant fields (as described above for *aspell*), and feed them into a character n-gram program to produce lists of n-grams of various lengths. Token counts on the various n-grams can then be used to find rare n-grams, which may be errors. Another approach would be to parse the input into syllables, although I have not tried this as yet.

References

Bird, Steven; and Gary Simons. 2003. "Seven dimensions of portability for language documentation and description". *Language* 79:557-582.

[7] Both *aspell* and the similar *ispell* program are freely available, and run under Linux or the CygWin environment under Windows, as well as coming in native Windows versions. There are dozens of language-particular dictionaries for *aspell* and *ispell*, see http://aspell.net/ and http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html.

[8] Again, *iconv* is freely available.

[9] A list of installed languages is available from Word's Language dialog box.

Leveraging the open source ispell codebase for minority language analysis

László Németh*, Viktor Trón†, Péter Halácsy*, András Kornai‡, András Rung *, István Szakadát*

*Budapest Institute of Technology Media Research and Education Center
{nemeth,halacsy,rung,szakadat}@mokk.bme.hu
†International Graduate College, Saarland University and University of Edinburgh, v.tron@ed.ac.uk
‡MetaCarta Inc., andras@kornai.com

Abstract

The *ispell* family of spellcheckers is perhaps the single most widely ported and deployed open-source language tool. Here we describe how the *SzóSzablya* 'WordSword' project leverages `ispell`'s Hungarian descendant, `HunSpell`, to create a whole set of related tools that tackle a wide range of low-level NLP-related tasks such as character set normalization, language detection, spellchecking, stemming, and morphological analysis.

1. Introduction

Over the years, open source unix distributions have become the definitive repositories of tried and tested algorithms. In the area of natural language processing, wellformedness of words is typically checked by the `ispell` family of spellcheckers that goes back to Gorin's `spell` program (see Peterson 1980), a spellchecker for English written in PDP-10 assembly. Since at the core of spellchecking is a method for accurate word recognition, it is an ideal platform both for reaching "down" toward language identification and for reaching "up" toward stemming and morphological analysis. The *SzóSzablya* 'WordSword' project at the Budapest Institute of Technology leverages the `ispell` methods with the goal to extend them to a general toolkit applicable to various low-level NLP-related problems other than spell-checking such as language detection, character set normalization, stemming, and morphological analysis.[1]

The algorithms described here go back to the roots of the `spell -- ispell -- International Ispell -- MySpell -- HunSpell` development. The linguistic theory implicit in much of the work has an even deeper historical lineage, going back at least to the Bloomfield–Bloch–Harris development of structuralist morphology via Antal's (1961) work on Hungarian. Despite our indebtedness to these traditions, this paper does not attempt to faithfully trace the twists and turns of the actual history of ideas, rather it offers only a rational reconstruction of the underlying logic.

A high performance spellchecker can easily be leveraged for language identification, and we have relied heavily on `HunSpell` both for this purpose and for overall quality improvement in creating a gigaword Hungarian corpus (see the main conference paper paper Halácsy et al 2004). Orthographic form and, by implication, spellchecking technology, remains the Archimedean point of natural language text processing both "downward" and "upward". Here we will concentrate entirely on the "upward" developments leading to `HunStem`, a full featured industrial strength stemmer that supports large-scale Information Retrieval applications, and eventually to `HunMorph`, an open source morphological analyzer.[2] Though the names `HunSpell` and `HunStem` suggest Hungarian orientation, in the spirit of `ispell` our project keeps the technology perfectly separated from lexical resources, making the tools are directly applicable to other languages provided that lexical databases are available. Resources for the applications can be compiled from a single lexical database and morphological grammar with the help of the `HunLex` resource compilation tool.

The paper is structured as follows. Section 2 provides a brief introduction to the morphological analysis/generation problem from the perspective of spellchecking, and discusses how the affix-flag mechanism introduced to `ispell` by Ackerman in 1978[3] has been modified to deal with multi-step affix stripping to attack the problem of languages with rich morphology. Section 3 describes how, by enabling multiple analyses, treatment of homonyms, and flexible output of stem information, the general framework of `HunSpell` has been extended to support stemming. In the concluding Section 4 we describe how the codebase can be leveraged even further, to support detailed morphological analysis.

2. The morphological OOV problem

The simplest spellchecker, both conceptually and in terms of optimal runtime performance, is a list of all correctly spelled words. Acceleration and error correction techniques based on hashes, tries, and finite automata have been extensively studied, and the implementor can choose from a variety of open source versions of these techniques. Therefore the spellchecking problem could be reduced, at least conceptually, to the problem of listing the correct words, whereby errors of the spellchecker are reduced to out of vocabulary (OOV) errors. A certain amount of OOV error is inevitable: new words are coined all the time, and the supply of exotic technical terms and proper names is inexhaustible. But as a practical matter, developers encounter

[1] Aversano et al 2002 is the only related attempt we know of.

[2] For further upward developments such as named entity extraction, parsing, or semantic analysis, orthography gradually loses its grip over the problem domain, but none of these higher-level developments are feasible without tackling the low-level issues first.

[3] For the history of `ispell/MySpell`, see the man pages.

OOV errors early on from another source: morphologically complex words such as compounds and affixed forms.

The ability to reverse compounding and affixation has a very direct payoff in terms of reducing memory footprint, and it is no surprise that affix stripping ability was built into `ispell` early on. Initially, `(i)spell` only used heuristics for affix stripping before looking up hypothesized stems in a base dictionary. This was substantially improved by the introduction of *switches* (in linguistic terms these would be called *privative lexical features*) that license particular affix rules and thus help eliminate spurious hits resulting from the unreliable heuristic method.

In 1988 Geoff Kuenning extended affix flags to license sets of affix rules. In this table-driven approach, affix flags are interpreted as lexical features that indicate morphological subparadigm membership. This method of affix compression allowed for less redundant storage and efficient run-time checking of a great number of affixes, thereby enabling `ispell` to tackle languages with more complex morphological systems than English. After major modifications of the code, the first multi-lingual version of `ispell` was released in 1988.

`Ispell` can also handle compounds and there is even the possibility of specifying lexical restrictions on compounding, also implemented as switches in the base dictionary. For some languages, a rich set of compound constructions allow for productive extensions of the base vocabulary, and this feature is indispensable in mitigating the OOV problem. Language-specific word-lists and affix rules for `ispell`, with added switch information as necessary, have been compiled for over 50 languages so far. Our development started with providing open source spell-checking for Hungarian. Our spellchecker, `HunSpell` is based on `MySpell`, a portable and thread-safe C++ library reimplementation of `ispell` written by Kevin Hendricks. We chose `MySpell` as the core engine for our development both because of its implementational virtues, and because the non-restrictive BSD license significantly enhances its potential in open source development and large-scale code reuse.[4]

The lexical resources of `MySpell` (the affix file and the dictionary file) are processed at runtime, which makes them directly portable across various platforms. A line in the affix file represents an affix rule from a generation point of view. It specifies a regexp-like pattern which is matched against the beginning or end of the base for prefix and suffix respectively, a string to strip from, and the actual affix string to append to the input base. A special indexing technique, the Dömölki algorithm is used in checking affixation conditions (Dömölki 1967) to pick applicable affix rules in parsing. A pseudo-stem is hypothesized by reverse application of the affix rule (i.e., stripping the append string and appending the strip string) which is looked up in the dictionary. A line in the dictionary file represents a lexical entry, i.e., a base form associated with a set of affix flags. If the hypothesized base is found in the dictionary after the application of an affix rule, in the last step it is checked whether its flags contain the one the affix rule is assigned to.

Though the ispell algorithm performs affix stripping and lexical lookup very efficiently, the implementation does not scale well to languages with rich morphology. `ispell` lexical resources actually exist for some languages with famously rich productive morphology such as Estonian, Finnish, and Hungarian, but it is suggestive that the latter two languages use enhancements over `MySpell` in their native OpenOffice.org releases for spellchecking.[5] The Hungarian version uses our development, the `HunSpell` library which incorporates various spell-checking features specifically needed to correctly handle Hungarian orthography – we turn to these now.

So far we spoke of affixes only in the sense of edge-aligned substrings (prefixes and suffixes), but in languages with complicated combinatorial morphology affix rules might stand for intricate clusters of affix morphemes (the sense of affix used in linguistics). Such morphotactic complexity, a hallmark of rich productive morphology, often makes it difficult to list all legitimate affix combinations, let alone produce them automatically: the sheer size and redundancy of precompiled morphologies make modifications very difficult and debugging nearly impossible. Maintaining these resources without a principled framework for off-line resource compilation is virtually a hopeless enterprise, witness `magyarispell`, the Hungarian `MySpell` resource[6] which resorts to a clever (from a maintainability perspective, way too clever) mix of shell scripts, m4 macros, and hand-written pieces of `MySpell` resources.

To remedy this problem we devised an off-line resource compilation tool which given a central lexical database and a morphological grammar can create resources for the applications according to a wide range of configurable parameters. `HunLex` is a language-independent pre-processing framework for a rule-based description of morphology (details about grammar specifications and configuration options of `HunLex` would go beyond this paper).

To handle all the productive inflections `magyarispell` requires about 20 thousand combined entries. Extending this database to incorporate productive derivational morphology would mean an order of magnitude increase, as full derivation and inflection can yield ca. 10^3-10^6 word forms from a single nominal base. Taking orthogonal prefix combinations into account would result in another order of magnitude increase, leading to file sizes unacceptable in a practical system.

Using the `magyarispell` resources, on the 5 million word types of the SzóSzablya web corpus (Halácsy et al. 2004), `HunSpell`'s recognition performance is about 96% (OOV is 4%). Taking word frequencies into account OOV is only 0.2% (i.e. recall is near perfect, 99.8%). While these figures are quite reassuring for the central use case of flagging spelling errors, in order to offer high quality replacements we can't ignore rare but perfectly well-formed complex forms. Decreased OOV is also indispensable for wide-coverage morphological analysis and Information Re-

[4] MySpell has been incorporated into OpenOffice.org's office suite, where it replaces the third-party libraries licensed earlier.

[5] The Finnish version is a closed-source licensed binary (see `http://www.hut.fi/ pry/soikko/openoffice/`).

[6] `http://magyarispell.sourceforge.net/`

trieval applications.

To solve the morphological OOV problem `HunSpell` now incorporates a multi-step sequential affix-stripping algorithm. After stripping an affix-cluster in step i, the resulting pseudo-stem can be stripped of affix-clusters in step $i + 1$. Legitimate strippings can be checked in exactly the same way as for valid online base+affix combinations, and are encoded with the help of switches in the resource file. Implementing this only required a minor extension of the data structure coding affix entries and a recursive call for stripping. Currently this scheme is implemented for two steps (plus lexical lookup) for suffixation plus one for prefixation, but can easily be extended to a fully recursive method.[7] From the structuralist perspective, the clustering step implements a kind of position class analysis (Nida 1949, Harris 1951), and from a generative perspective it implements a simplified version of lexical phonology and morphology (Kiparsky 1982). Besides the well-known theoretical justifications for this style of analysis, there is a compelling practical justification in that the size of the affix table shrinks substantially: with our particular setting for Hungarian, the multi-step resource is the square root of the single-step one in size. `HunLex` can be configured to cluster any or no set of affixes together on various levels, and therefore resources can be optimized on either speed (toward one-level) or memory use (affix-by-affix stripping).

Prefix–suffix dependencies An interesting side-effect of multi-step stripping is that the appropriate treatment of circumfixes now comes for free. For instance, in Hungarian, superlatives are formed by simultaneous prefixation of *leg-* and suffixation of *-bb* to the adjective base. A problem with the one-level architecture is that there is no way to render lexical licensing of particular prefixes and suffixes interdependent, and therefore incorrect forms are recognized as valid, i.e. **legvén* = *leg* + *vén* 'old'. Until the introduction of clusters a special treatment of the superlative had to be hardwired in the earlier `HunSpell` code. This may have been legitimate for a single case, but in fact prefix–suffix dependences are ubiquitous in category-changing derivational patterns (cf. English *payable*, *non-payable* but **non-pay* or *drinkable*, *undrinkable* but **un-drink*). In simple words, here, the prefix *un-* is legitimate only if the base *drink* is suffixed with *-able*. If both these patters are handled by on-line affix rules and affix rules are checked against the base only, there is no way to express this dependency and the system will necessarily over- or undergenerate.

Compounds Allowing free compounding yields decrease in precision of recognition, not to mention stemming and morphological analysis. Although lexical switches are introduced to license compounding of bases by `ispell`, this proves not to be restrictive enough. This has been improved upon with the introduction of direction-sensitive compounding, i.e., lexical features can specify separately whether a base can occur as leftmost or rightmost constituent in compounds. This, however, is still insufficient to handle the intricate patterns of compounding, not to mention idiosyncratic (and language specific) norms of hyphenation.

The `MySpell` algorithm currently allows any affixed form of words which are lexically marked as potential members of compounds. `Hunspell` improved upon this, and its recursive compound checking rules makes it possible to implement the intricate spelling conventions of Hungarian compounds. This solution is still not ideal, however, and will be replaced by a pattern-based compound-checking algorithm which is closely integrated with input buffer tokenization. Patterns describing compounds come as a separate input resource that can refer to high-level properties of constituent parts (e.g. the number of syllables, affix flags, and containment of hyphens). The patterns are matched against potential segmentations of compounds to assess wellformedness.

3. Stemming and morphological analysis

So far, we only touched upon general issues pertaining to the recognition of morphologically complex forms in highly inflecting languages. It is easy to realize, however, that the same general architecture can easily be extended to more sophisticated analysis tools for morphological processing. A straightforward extension we implemented allowed `HunSpell` to output lexical stems, thereby turning it into a simplistic stemmer.

Practically, stemmers are used as a recall enhancing device for Information Retrieval systems (Kraaij and Pohlmann 1996, Hull 1996). Stemmers ideally conflate semantically related wordforms, so indexing words by their stems effectively expands the relevant search space. The relevance of this ubiquitous NLP technique is greater for languages with rich (inflectional) morphology and/or relatively smaller corpus. Stemmers based on various approximate heuristics (Porter 1980, Paice 1994) are already quite robust and ones based on corpus statistics can be totally language independent (Xu and Croft 1998). However, these methods very often produce nonwords the human interpretation of which is difficult, which makes debugging of false conflations hard. Therefore, once linguistic resources are available, stemming based on linguistically motivated morphological analysis is beneficial at least from a maintainability perspective.

To turn `HunSpell` into `HunStem`, a fully functional grammar-based stemmer, we had to address several issues beyond the trivial provision for stem output. First, for the recognition problem relevant in word-based spellchecking, no multiple analyses are needed, so the processing of a word can terminate with the first successful analysis. For any stemmer of practical use, this is insufficient, and coming up with alternative stems for morphologically ambiguous forms is a definitive requirement. This has been implemented and `HunStem` now performs exhaustive search for analyses and outputs all potential stems.

Second, for stemming it is desirable that homonymous stems be disambiguated if affixation provides the necessary cues. This is usually the case with ambiguous stems belonging to different paradigms or categories like Hun-

[7] For the fully recursive version, in order to guarantee termination, one has to either impose a limit on the number of steps or make sure that the lengths of pseudo-stems that are the result of successive legitimate stripping operations converge to zero.

garian *hal*, which is ambiguous between the verbal reading 'die' and the nominal reading 'fish'. In the original design, string-identical bases are conflated and there is no way to tell them apart once their switch-set licensing various affixes are merged. Fixing this only required minor technical modifications in the code and `HunStem` is now able to handle homonyms and output the correct stem if the base occurs in disambiguating affix pattern. Note that base-licensing of incompatible affixes for homonymous stems is a problem for recognition as well. For instance, in Hungarian, *hal*, used as a verb, can take various verbal affixes, but these cannot cooccur with nominal affixes. The problem is that the architecture is unable to rule out homonymous bases with illegitimate simultaneous cross-category combinations of prefix and suffix such as **meghalam* = *meg* verbal prefix + *hal* 'die V' + *-am* 'POSS_1SG nominal suffix'. Before the correct treatment of homonymous bases was introduced, the only available solution was to list precompiled verbal and nominal paradigms separately for these, which is not only wasteful and error-prone, but also puts productively derived forms out of scope.

Third, and most importantly, the literal output of lexical stems looked up in the dictionary resource after affix stripping may not be optimal for stemming purposes without explicitly addressing the issue of when to terminate the analysis. From the perspective of spellchecking there was no reason to get rid of exactly the affixes one is likely to want to strip in a typical stemming task. Since the division of labor between online (runtime) and offline (compile time) affixation is irrelevant for the recognition task, choices are mainly biased to optimize efficient storage and/or processing rather than to reflect some meaningful coherence of dictionary bases. That is, several morphologically complex forms may be appear as bases (either listed or off-line precompiled) in the original `HunSpell` resource dictionary, which was not optimized for stemming output. For instance, Hungarian exceptional singular accusative *farkat* is linguistically derived from stem *farok* but the ispell analysis was based on a dictionary entry for the plural nominative *farkak* for reasons of efficient coding. In the current system the adjustment of conflation classes, i.e., which words are kept unanalyzed and which affixes are stripped, is therefore flexibly configurable: the precompiler `HunLex`, which replaces our earlier set of `m4` macros, creates lexical resources for the stemmer based on various parameters, which opens the door to the creation of task-dependent stemmers optimized differently for different IR applications.

4. Conclusion

If we aspire to scale open source language technology to a wide range of languages, the problems exemplified by Hungarian are but instances of the general problems one will necessarily encounter along the way, because a substantial proportion of the world's languages (e.g., Altaic, Uralic and Native American languages) are heavily agglutinative. Since scaling to other languages is an important motivation behind developing our toolkit, we believe that even those languages with rich morphology, like Turkish or Basque, which as yet lack MySpell lexical resources, will eventually benefit from our efforts.

The next logical step is a full-fledged morphological analysis tool for Hungarian. Many of the prerequisites of morphological analysis, in particular the flexibility to define the set of morphemes left unanalyzed at compile time, were fulfilled in the course of the `HunStem` development, and a pilot version of `HunMorph` is already operational. In principle, `HunLex` method of dictionary resource precompilation is applicable even to Kimmo-style systems, where the inner loop is based on finite state transduction rather than the generic string manipulation techniques used in `ispell`, but in the absence of a non-restrictive license open source two-level compiler we are not in a position to pursue this line of research.

Acknowledgements

The SzóSzablya project is funded by an ITEM grant from the Hungarian Ministry of Informatics and Telecommunications, and benefits greatly from logistic and infrastructural support of MATÁV Rt. and Axelero Internet.

5. References

L. Antal, 1961. A magyar esetrendszer [The Hungarian case system]. *Nyelvtudományi Értekezések*, 29:57–77.

L. Aversano, G. Canfora, A. De Lucia, and S. Stefanucci. 2002. Evolving ispell: A case study of program understanding for reuse. In *Proceedings of the 10th International Workshop on Program Comprehension*, p197. IEEE Computer Society.

B. Dömölki. 1967. Algorithms for the recognition of properties of sequences of symbols. *USSR Computational & Mathematical Physics*, 5(1):101–130. Pergamon Press, Oxford.

P. Halácsy, A. Kornai, L. Németh, A. Rung, I. Szakadát and V. Trón 2003. Creating open language resources for Hungarian. See LREC Proceedings.

Z. Harris. 1951. *Methods in Structural Linguistics*. University of Chicago Press, Chicago.

D. A. Hull. 1996. Stemming algorithms: a case study for detailed evaluation. *J. Am. Soc. Inf. Sci.*, 47(1):70–84.

C. Jacquemin. 1997. Guessing morphology from terms and corpora. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, 156–165. ACM Press.

P. Kiparsky. 1982. Lexical phonology and morphology. In I.S. Yang, editor, *Linguistics in the Morning Calm*, 3–91. Hansin, Seoul.

W. Kraaij and R. Pohlmann. 1996. Viewing stemming as recall enhancement. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 40–48. ACM Press.

J. B. Lovins. 1968. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

E. Nida. 1949. *Morphology: The Descriptive Analysis of Words*. University of Michigan, Ann Arbor.

C. D. Paice. 1994. An evaluation method for stemming algorithms. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 42–50. Springer-Verlag New York, Inc.

J. L. Peterson. 1980. *Computer programs for spelling correction: an experiment in program design*, volume 96.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

J. Xu and W. B. Croft. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst.*, 16(1):61–81.

Analysing Irish prosody: A dual linguistic/quantitative approach

Ailbhe Ní Chasaide, Martha Dalton, Mika Ito, Christer Gobl

Phonetics and Speech Lab
Centre for Language and Communication Studies, Trinity College Dublin
anichsid@tcd.ie, daltonm@tcd.ie, mito@tcd.ie, cegobl@tcd.ie

Abstract

A project of Irish prosody is described which attempts to provide not only the basis for a linguistic description of the prosody of Irish dialects, but also the prerequisite quantitative characterization that is needed to allow us to use it for future technological applications, particularly text-to-speech development for Irish dialects. As with many other minority languages, there are particular challenges, but also particular opportunities to address. A multi-layered analytic approach is adopted, which will provide coverage of the three phonetic dimensions of prosody: pitch dynamics (intonation); voice quality; and temporal features. It is also envisaged that these analyses will provide the basis for an account that encompasses both the narrowly linguistic functions of prosody and its paralinguistic function of signaling attitude and emotion. In these last respects, this study aims also to contribute to the broader understanding of prosody, and to its modeling for more expressive speech synthesis. Given the relatively threatened status of Irish, we hope that by gearing our linguistic analysis to eventual technology exploitation, we can go beyond the mere documentation and aspire to the provision of tools that can support language teaching/learning and language usage generally.

Introduction

In this paper we describe a new project on the Prosody of Irish Dialects. Linguistic research on a minority language, in comparison to widely spoken and widely analyzed languages, often presents particular challenges and opportunities. As the approach adopted reflects many of these concerns it may thus be of general interest.

In our group we are keenly aware that knowledge concerning the linguistic structure of a language is a prerequisite for many speech technology applications. For example, in order to develop even a basic text-to-speech system, we ideally need models of the prosodic structure, as well as an understanding of segmental and grammatical structure etc. However, the analyses are often ill adapted to the eventual technological exploitation, and there is often a gap than can not easily be bridged. To be truly useful for technology, we need to adopt methodologies that ensure that the output can be harnessed by technology.

In the case of minority languages, speech technology is not just a luxury or gimmick. For endangered languages in particular, linguists are increasingly aware of the need to record and document them for posterity. Tools such as high quality text-to-speech facilities would serve to preserve models of the spoken language. Beyond the 'preserving' function, they could further play an important role in supporting language teaching/learning and language usage, crucial to the survival of the language.

The conundrum is that the non-commercial status of these languages makes it difficult for them to attract the funding and manpower for such linguistic research and technology adaptations. The lack of commercial viability arises from the limited size of the community of users, and/or as is the case with Irish, from the fact that most users are bilingual, so that product producers see the market as already served.

The need for Irish prosody research

Many of the goals of the present project have been shaped by these kinds of considerations, and it is felt that the linguistic analysis must maximize the potential downstream usefulness for later technology applications, particularly text-to-speech development. It therefore relates closely to another ongoing project, WISPR, which directly targets such technological developments (see parallel paper in this Workshop).

In setting out to describe Irish prosody, there are some further challenges which are shared with many other minority languages. Firstly, there is no prior account to draw on. Although there have been many accounts of the segmentals of Irish dialects (de Búrca, 1958; de Bhaldraithe, 1945), there is to date virtually no available coverage of segmental aspects, other than some short fragments on intonation (Blankenhorn, 1982; de Bhaldraithe, 1945). Consequently, the research is necessarily an exploration of mostly unchartered territory.

Another issue, common to many minority languages, is that given the historical and sociological context, there is no standard dialect. Rather, for Irish, we are confronted with four main dialects: and focusing on a single one involves making assumptions as to which one we regard as dominant. For that reason, in the present project, a cross-dialect approach has been adopted, and the analysis will be carried out on the four dialects of Donegal, Mayo, Connemara and Kerry Irish as illustrated in Figure 1.

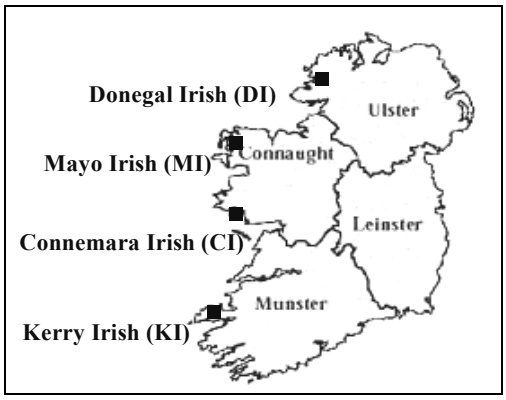


Figure 1: Map showing location of the four dialects of Irish, included in the analysis

One of the major linguistic interests of the descriptive material will be to show up what is common to all dialects and where the major divergences occur. The cross-dialect approach is dictated also partly by the fact that we are eventually interested in the provision of text-to-speech for all the main dialects.

It should be pointed out that research in minority languages also offers particular opportunities. First of all it allows us to reconsider many assumptions about language structure universals which are based on analyses of very few languages, mostly English. It is also an opportunity to look in a fresh way at theoretical issues and methodologies, and to try new perspectives, as one is in some sense "freed up" from current methodological straight jackets.

Project goals

As an approach to prosody description this project adopts a number of strategies, some of which are in line with current linguistic analyses, some of which are rather novel. The specific goals and methodologies are where relevant tailored to the situation of Irish and our hopes for technological downstream dividends. They also reflect some of the ongoing research preoccupations of our laboratory, and an interest in providing a more holistic account of prosody in general, which of course would have implications for the broader field of synthesis of any language. From this perspective, there are three main goals.

1. Integrating the three phonetic dimensions

It will describe the three phonetic dimensions of prosody: intonation, voice quality and rhythmic/temporal features. While most descriptive analyses of the prosody of individual languages provide an account of melodic structure, we feel that all three (in our view) phonetic correlates need to be incorporated where possible. Thus we will aim to provide an account of:

Intonation/melody: account of the primary pitch patterns of each dialect, and how these differ.

Voice quality: account of voice source correlates of major intonational categories, and an account of how voice quality is exploited for linguistic differentiation and for paralinguistic communication of affect and attitude (see next subsection). This is largely absent from descriptions of prosody, but some of our own research (e.g. Ní Chasaide and Gobl 2004) as well as the work of other researchers in the field indicate that coverage of pitch dynamics without taking corresponding modulation of the voice source into account is inevitably a partial treatment of prosodic phenomena. This for us is very linked to some basic research on the voice which we are carrying out in the lab. It is also of course relevant to downstream synthesis applications, although this is in the longer term.

In technical terms, this is a difficult area to work in as obtaining reliable measurements is not at all straightforward. For this reason, voice quality analysis will be limited to two of the dialects.

Temporal/durational features: a description of the salient rhythmic/temporal characteristics of the dialects.

Although timing and rhythm is a major area of prosodic research, there is often a divide between it and that research focused on melody. To gain a proper understanding of how the prosody of a language works, we feel that a parallel description is required, so that we get some sense of how the temporal features contribute to those prosodic elements that are often described as involving pitch alone.

The need to include an account of temporal and rhythmic factors arises also from our long-term perspective of providing prosodic modeling for speech synthesis of Irish. Finally, within Irish linguistics, much of the interest will be focused on cross-dialect divergence and on the rather striking rhythmic differences in these dialects.

2. Encompassing linguistic/paralinguistic functions.

A second major goal is to provide an account that bridges the gap between the linguistic and paralinguistic functions of prosody. Early treatments of intonation such as O'Connor and Arnold (1961) saw the paralinguistic signaling of attitude and emotion as a primary role of prosody in speech communication, more recent generations of linguists have tended to shy away from this aspect, and focus more narrowly on its linguistic functions such as marking focus, phrase boundaries or even sentence mode (differentiation between declaratives and interrogatives).

While the initial thrust of our analyses will also target the linguistic level, we will be incorporating some analyses of affectively colored speech with a view to providing some initial model of how the linguistically relevant constituents of prosody vary with the attitude and emotion of the speaker.

This particular aspect of the project links to research we have been doing on the mapping of different voice qualities to affect (Gobl and Ní Chasaide, 2003) and on how voice quality and f0 combine in affect signaling (Gobl et al 2002, Ní Chasaide and Gobl 2004). It is our firm view that the field of prosody has become falsely fragmented in a number of ways, and that we need to look at linguistic and paralinguistic phenomena within a single

framework. This is of course related to our intention of drawing together the different phonetic correlates of prosody: it is unlikely that one can make progress in describing the paralinguistic dimension without incorporating voice quality, along with pitch dynamics and temporal aspects.

Our research to date on voice quality, pitch and affect signaling was carried out through perception experimentation (references as above) with stimuli generated using formant based synthesis and a sophisticated voice source model (Fant et al., 1985). Our present project provide us with an opportunity to combine this with an analytic approach. This aspect of our research links to the objectives of a newly launched network of excellence on emotion (HUMAINE, 2004).

It also relates to possible downstream developments in speech synthesis, and in particular the provision of synthesized voices which are capable of conjuring specific emotions and attitudes (Gobl et al, 2002; Gobl and Ní Chasaide, 2002; Ní Chasaide and Gobl, 2002).

3. Qualitative and quantitative coverage

In keeping with the fact that we as linguists want to provide a linguistic account which is readily harnessed for technology applications, we are adopting a dual qualitative and quantitative approach. In the first, current phase, a linguistic, qualitative analysis of corpus materials is being carried out within the framework of Autosegmental-Metrical phonology (Ladd, 1996), using tone labels adapted from the ToBI annotation system. This analysis is an account of the possible combinations of tones (e.g. H*, L*+H) associated with accented syllables and phrase boundaries. The analysis is essentially auditory, and the researcher is guided by the visual display of f0. Figure 2 illustrates a ToBI-type analysis with f0 display for the sentence *Bíonn ealaí ag snámh in Árainn Mhór*. This is the current standard for intonational analysis, and an account in this framework will be broadly accessible and allow comparison with other studies on languages and dialects elsewhere.

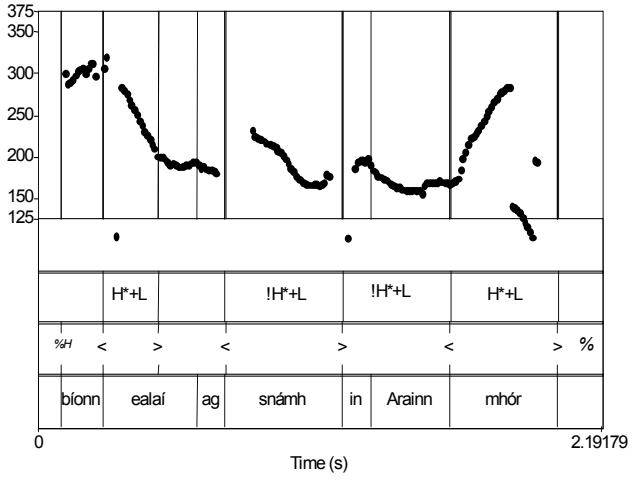


Figure 2: Display of Autosegmental-Metrical analysis of an Irish utterance, showing from top: f0 contour; tonal labels; prominence and boundary labels; and orthographic representation.

In the second phase of our intonation analysis, the same materials will be quantitatively analysed, using the Fujisaki model (Fujisaki, 1983). Figure 3 illustrates the same sentence, whose intonation is modeled in terms of Fujisaki parameters.

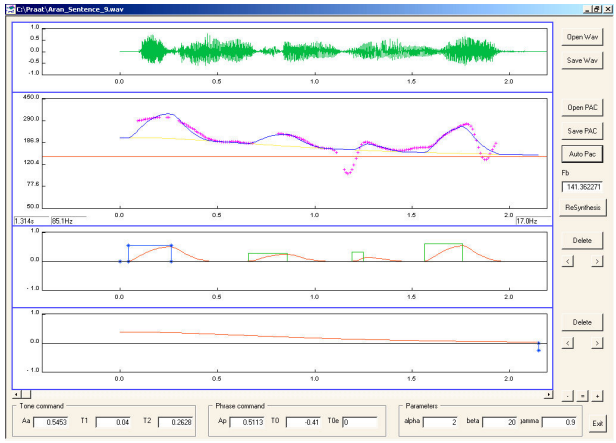


Figure 3: Display of Fujisaki analysis of the Irish utterance in Figure 2, showing from top: speech waveform; the original f0 contour (*dotted line); the modelled Fujisaki contour (solid line); the Fujisaki accent commands; and the Fujisaki phrase command.

The Fujisaki modeling is well adapted to synthesis applications and to our technology related goals for Irish. The quantitative measurements are also required by the second goal mentioned above, namely that of providing coverage of paralinguistic and linguistic phenomena within a single framework. As regards intonation, many of the prosodic manifestations of attitude and emotion concern not the configuration of tonal sequences, but rather changes in the dynamic range and the average level of f0 (Scherer, 1986). The Autosegmental-Metrical approach, using ToBI-type labels is highly abstract. While it is admirably suited to capture the linguistic, contrastive aspects of intonation, for which it was devised, it abstracts away much of the paralinguistic related variation.

Conclusion

In this project our goals and analytic strategies are motivated by different though not necessarily competing considerations:

- providing an account of an unchartered aspect of linguistic structure;
- providing an account of Irish intonation that is accessible to those interested in the typology of intonation systems;
- providing quantitative modeling of intonation and temporal structure that we can subsequently exploit in developing technology applications, particularly in the provision of text-to-speech for Irish
- providing basic research that will contribute towards a better understanding of the nature of prosody, and in particular of how the same set of phonetic dimensions (pitch, voice quality, and timing) are simultaneously exploited by the speaker to convey both linguistic and paralinguistic information.

To the extent that this last objective is achieved, this work will be contributing also to the development of more human-like expressive synthetic voices.

Acknowledgements

This research has been financially supported by a Government of Ireland Senior Research Fellowship to the first author, funded by the Irish Research Council for Research in the Humanities and Social Sciences, and by the research project *Prosody of Irish Dialects: the use of intonation, rhythm and voice quality for linguistic and paralinguistic signalling*, which is also funded by the Irish Research Council for Research in the Humanities and Social Sciences.

References

Blankenhorn, V. S. (1982) Intonation in Connemara Irish: A Preliminary Study of Kinetic Glides. In Studia Celtica 16-17 (pp. 259-279).

de Bhaldraithe, T. (1945) The Irish of Cois Fhairrge, Co. Galway, Dublin Institute for Advanced Studies.

de Búrca, S. (1958) The Irish of Tourmakeady, Co. Mayo. Dublin Institute for Advanced Studies.

Fant, G., Liljencrants, J. and Lin, Q. (1985). A four-parameter model of glottal flow. *STL-QPSR*, Speech, Music and Hearing, Royal Institute of Technology, Stockholm, 4, 1-13.

Fujisaki, H (1983). Dynamic characteristics of voice fundamental frequency in speech and singing. In Peter F. MacNeilage (Ed), The production of speech (pp. 39–55). Springer, New York.

Gobl, C. and Ní Chasaide, A. (2002). Dynamics of the Glottal Source Signal: Implications for Naturalness in Speech Synthesis. In E. Keller, G. Bailly, A. Monaghan, J. Terken and M. Huckvale (Eds.) Improvements in Speech Synthesis (pp. 273-283), Wiley and Sons.

Gobl, C. and Ní Chasaide, A. (2003). The role of voice quality in communicating emotion, mood and attitude. *Speech Communication*, 40, 189-212.

Gobl, C., Bennett, E. and Ní Chasaide, A. (2002). Expressive synthesis: how crucial is voice quality. In Proceedings of the IEEE Workshop on Speech Synthesis, Santa Monica, California, paper 52, 1-4.

HUMAINE Network of Excellence (2004-2008). Funded by the EU Sixth Framework Programme. http://emotion-research.net

Ladd, D. Robert (1996). Intonational Phonology. Cambridge University Press

Ní Chasaide, A. and Gobl, C. (2002). Voice Quality and the Synthesis of Affect. In E. Keller, G. Bailly, A. Monaghan, J. Terken and M. Huckvale (Eds.) Improvements in Speech Synthesis, Wiley and Sons, 252-263.

Ní Chasaide, A. and Gobl, C. (2004). Voice quality and f_0 in prosody: towards a holistic account. *Proceedings of the 2nd International Conference on Speech Prosody*, Nara, Japan.

O'Connor, J.D. and Arnold, G.F. (1961). The Intonation of Colloquial English. Longman, London.

Scherer, K.R. (1986). Vocal affect expression: A review and a model for future research. Psychological Bulletin, 99, 143-165.

Creating a Morphological Analyzer and Generator for the Komi language

Attila Novák

MorphoLogic Ltd.
Budapest, Orbánhegyi út 5., 1126 Hungary
novak@morphologic.hu

Abstract

In this paper a set of tools for morphological analysis and generation is presented along with its application to Komi-Zyryan, a small Finno-Ugric language spoken in Northeastern Europe. This endeavor is part of a project which aims to create annotated corpora and other electronically available linguistic resources for a number of small members of the Uralic language family. A morphological grammar development environment is also introduced which facilitates a rapid development of the morphological descriptions used by the tools presented.

1. Introduction

Various Hungarian research groups specialized in Finno-Ugric linguistics and a Hungarian language technology company, MorphoLogic have initiated a project with the goal of producing annotated electronic corpora for small Uralic languages. This paper describes the current state of the subproject on Komi.

2. The Tools

In the project, we use a morphological analyzer engine called Humor ('High speed Unification MORphology') developed at MorphoLogic (Prószéky and Kis, 1999). We supplemented the analyzer with two additional tools: a lemmatizer and a morphological generator.

2.1. The Morphological Analyzer

Features of the Analyzer The Humor analyzer performs a classical 'item-and-arrangement' (IA) style analysis (Hockett, 1954). The input word is analyzed as a sequence of morphs. It is segmented into parts which have (i) *a surface form* (that appears as part of the input string), (ii) *a lexical form* (the 'quotation form' of the morpheme) and (iii) *a category label* (which may contain some structured information or simply be an unstructured label). The lexical form and the category label together more or less well identify the morpheme of which the surface form is an allomorph.

Although the 'item-and-arrangement' approach to morphology has been criticized, mainly on theoretical grounds, by a number of authors (c.f. e.g. Hockett, 1954; Hoeksema and Janda, 1988; Matthews, 1991), the Humor formalism had been in practice successfully applied to languages like Hungarian, Polish, German, Rumanian and Spanish so we decided to use it in this project as well. The 'slicing-up' approach of the analyzer we use seemed suitable to the agglutinating type of languages to which Komi belongs. On the other hand, we avoided segmenting any 'portemanteau' morphemes the segmentation of which would have been purely stipulated.

Another feature of the analyzer is that it produces flat morph lists as possible analyses, i.e. it does not assign any internal constituent structure to the words it analyzes. The reason for this is that it contains a regular word grammar, which is represented as a finite-state automaton. This is clearly much more efficient than having a context-free parser and it also avoids most of the irrelevant ambiguities a CF parser would produce.

The following is a sample output of the Humor analyzer for the Komi word form *kolanla*.

```
analyzer>kolanla
kov[S_V]=kol+an[D=A_PImpPs]+la[I_CNS]
kov[S_V]=kol+an[D=N_Tool]+la[I_CNS]
```

Morphs are separated by + signs from each other. The representation morphs is `lexical form[category label]=surface form`. The surface form is output only if it differs from the lexical form. A prefix in category labels identifies the morphological category of the morpheme (`S`: stem, `D`: derivational suffix, `I`: inflectional suffix). In the case of derivational affixes, the syntactic category of the derived word is also given.

In the example above, *kov* is identified as the lexical form of a verb stem (`S_V`). The stem undergoes a stem alternation the result of which is that its surface form end in *-l* instead of *-v*. A derivational suffix *-an* is attached to it, the surface and lexical form of which is identical. The morph is ambiguous: it is either a noun forming suffix or a suffix forming a passive participle. This is followed by an inflectional suffix: the consecutive case marker *-la*.

How the analyzer works The program performs a search on the input word form for possible analyses. It looks up morphs in the lexicon the surface form of which matches the beginning of the input word (and later the beginning of the yet unanalyzed part of it). The lexicon may contain not only single morphs but also morph sequences. These are ready-made analyses for irregular forms of stems or suffix sequences, which can thus be identified by the analyzer in a single step, which makes its operation more efficient.

In addition to assuring that the requirement that the surface form of the next morpheme must match the beginning of the yet unanalyzed part of the word (uppercase-lowercase conversions may be possible) is met, two kinds of checks are performed by the analyzer at every step, which make an early pruning of the search space possible.

On the one hand, it is checked whether the morph being considered as the next one is locally compatible with the previous one. On the other hand, it is examined whether the candidate morph is of a category which, together with the already analyzed part, is the beginning of a possible word construction in the given language (e.g. suffixes may not appear as the first morph of a word etc.). The global word structure check is performed on candidate morphs

for which the local compatibility check has succeeded. Possible word structures are described by an extended finite-state automaton (EFSA) in the Humor analyzer.

2.2. The Lemmatizer

The lemmatizer tool, built around the analyzer core, does more than just identifying lemmas of word forms: it also identifies the exposed morphosyntactic features. In contrast to the more verbose analyses produced by the core analyzer, compound members and derivational suffixes do not appear as independent items in the output of the lemmatizer, so the internal structure of words is not revealed. [1] The analyses produced by the lemmatizer are well suited for such tasks as corpus tagging, indexing and parsing. The output of the lemmatizer and the analyzer is compared in the example below:

```
analyzer>kolanla
kov[S_V]=kol+an[D=A_PImpPs]+la[I_CNS]
kov[S_V]=kol+an[D=N_Tool]+la[I_CNS]

lemmatizer>kolanla
kolan[N][CNS]
kolan[A][CNS]
```

The lemmatizer identifies the word form *kolanla* as the consecutive case of the noun or adjective (in fact: participle) *kolan*.

2.3. The Generator

Originally, MorphoLogic did not have a morphological generator, so another new tool we created using the analyzer engine was a morphological generator. The generator produces all word forms that could be realizations of a given morpheme sequence. The input for the generator is normally very much like the output of the lemmatizer: a lemma followed by a sequence of category labels which express the morphosyntactic features the word form should expose.
The generator is not a simple inverse of the corresponding analyzer, thus it can generate the inflected and derived forms of any multiply derived and/or compound stem without explicitly referring to compound boundaries and derivational suffixes in the input even if the whole complex stem is not in the lexicon of the analyzer.
The following examples show how the generator produces an inflected form of the derived nominal stem *kolan*, which is not part of the stem lexicon, and the explicit application of the derivational suffix (and the same inflectional suffix) to the absolute verbal root of the word.

```
generator>kolan[N][CNS]
kolanla
generator>kov[V][_Tool][CNS]
kolanla
```

The development environment also makes it possible for the linguist to describe preferences for the cases when a certain set of morphosyntactic features may have more than one possible realization. This can be useful for such applications of the generator as text generation in machine translation applications, where a single word form must be generated.
We also created a version of the generator which accepts the morphosyntactic category labels in any order (as if it were just an unordered set of morphosyntactic features) and produces the corresponding word forms.

3. The Morphological Database

Various versions of the Humor morphological analyzer have been in use for over a decade now. Although the analyzer itself proved to be an efficient tool, the format of the original database turned out to be problematic. The operations that the analyzer uses when analyzing the input word must be very simple so that processing could be very efficient. This requires that the data structures it uses contain redundant data (so that they do not have to be calculated on the fly during analysis).
The most important problem with the Humor analyzer was that MorphoLogic had no tools for creating and maintaining these redundant data structures, which were hard to read for humans, and to modify in a consistent way. This resulted in errors and inconsistencies in the descriptions, which were difficult to find and correct.

3.1. Creating a Morphological Description

So the first thing to do was to create an environment which facilitates the creation of the database. In the new environment, the linguist has to create a high level human readable description which contains no redundant information and the system transforms it in a consistent way to the redundant representations which the analyzer uses. The work of the linguist consists of the following tasks:
Identification of the relevant morpheme categories in the language to be described (parts of speech, affix categories).
Description of stem and suffix alternations: an operation must be described which produces each allomorph from the lexical form of the morpheme for each phonological allomorphy class. The morphs or phonological or phonotactic properties which condition the given alternation must be identified.
Identification of features: all features playing a role in the morphology of the language must be identified. These can be of various sorts: they can pertain to the category of the morpheme, to morphologically relevant properties of the shape of a given allomorph, to the idiosyncratic allomorphies triggered by the morpheme or to more than one of these at the same time.
Definition of selectional restrictions between adjacent morphs: selectional restrictions are described in terms of requirements that must be satisfied by the set of properties (features) of any morph adjacent to a morph. Each morph has two sets of properties: one can be seen by morphs adjacent to the left and the other by morphs adjacent to the right. Likewise, any morph can constrain its possible neighbors by defining a formula expressing its requirements on each of its two sides.
Identification of implicational relations between properties of allomorphs and morphemes: these implicational relations must be formulated as rules, which define how redundant properties and requirements of allomorphs can be inferred from their already known (lexically given or previously inferred) properties (including their shape). Rules may also define default properties. relatively simple spe-

[1] The output of our lemmatizer is what is usually expected of a morphological analyzer.

cial-purpose procedural language, which we devised for this task, can be used to define the rules and the patterns producing stem and affix allomorphs.

Creation of stem and affix morpheme lexicons: in contrast to the lexicon used by the morphological analyzer, the lexicons created by the linguist contain the descriptions of morphemes instead of allomorphs. Morphemes are defined by listing their lexical form, category and all unpredictable features and requirements. Irregular affixed forms and suppletive allomorphs should also be listed in the lexicon instead of using very restricted rules to produce them. We implemented a simple inheritance mechanism to facilitate the consistent treatment of complex lexical entries (primarily compounds). Such items inherit the properties of their final element by default.

Creation of a word grammar: restrictions on the internal morphological structure of words (including selectional restrictions between nonadjacent morphemes) are described by the word grammar. The development environment facilitates the creation of the word grammar automaton by providing a powerful macroing facility.

Creation of a suffix grammar (optional): an optional suffix grammar can be defined as a directed graph, and the development environment can produce segmented suffix sequences using this description and the suffix lexicon. Using such preprocessed segmented sequences enhances the performance of the analyzer.

As it can be seen from the description of the tasks above, we encourage the linguist to create a real analysis of the data (within the limits of the model that we provide) instead of just blindly describing each word as one which belongs e.g. to class X23b.

3.2. Conversion of the Morphological Database

Using a description that consists of the information described above, the development environment can produce a lexical representation which already explicitly contains all the allomorphs of each morpheme along with all the properties and requirements of each allomorph. This representation still contains the formulae expressing properties and selectional restrictions in a human-readable form and can thus be easily checked by a linguist. The example below shows a representation of the alternating noun stem *lov* and the plural + second person plural possessive + consecutive case suffix sequence *jasnydla* from the Komi description.

```
lemma:  'lov[N]'
 form:  'lov'
  mcat: 'S_N'
  rp:   'cat_N sfxable mcat_stem'
  rr:   '!V_ini'
 form:  'lol'
  mcat: 'S_N'
  rp:   'cat_N sfxable mcat_stem'
  rr:   'V_ini'

lemma:  'jas[Pl]nyd[PSP2]la[CNS]'
 form:  'jas+nyd+la'
  mcat: 'I_Pl+I_PSP2+I_CNS'
  rp:   'mcat_infl'
  lp:   'Pl'
  lr:   'cat_Nom sfxable'
```

The noun (`N`) stem (`S_`) *lov* has two forms (allomorphs): *lov* and *lol*. Their right-hand side properties (`rp`) are: `cat_N` (syntactic category is noun), `sfxable` (suffixes may be attached) and `mcat_stem` (morphological category is stem). The allomorph *lov* also requires (`rr`) that the following morph should not be vowel-initial, while *lol* requires it to be vowel-initial.

The representation of the inflectional (`I_`) suffix sequence *jasnydla* states that it is composed of the surface forms *jas*, *nyd* and *la*, the category labels of which are `Pl` (plural), `PSP2` (possessive second person plural) and `CNS` (consecutive case), respectively. The properties of this form is `mcat_infl` (morphological category is inflection) and `Pl` (the first member of the sequence is a plural suffix). Its left neighbor must be a morph of a nominal category to which suffixes can be attached.

This representation is then transformed to the format used by the analyzer using an encoding definition description, which defines how each of the features should be encoded for the analyzer. The development environment makes it possible to express that certain properties are in fact mutually exclusive possible values of the same feature (eg. `cat_N` and `cat_V`) by decomposing them to independent binary properties in the encoding definition.

4. The Komi Analyzer

In the subproject on Komi, which concentrates on the standard Komi-Zyryan dialect, we created a Komi morphological description using the development environment described in the previous section. As a result, a working morphological analyzer, a lemmatizer and a generator have been produced.

4.1. The Language

Komi (or Zyryan, Komi-Zyryan) is a Finno-Ugric language spoken in the northeastern part of Europe, West of the Ural Mountains. The number of speakers is about 300 000. Komi has a very closely related language, Komi-Permyak (or Permyak, about 150 000 speakers), which is often called a dialect, but with a standard of its own.

As a language spoken in Russia, Komi is an endangered language. Although it has an official status in the Komi Republic (Komi Respublika), this means hardly anything in practice. The education is in Russian, children attend only a few classes in their mother tongue. A hundred years ago, 93% of the inhabitants of the region were of Komi nationality. Thanks to the artificially generated immigration (industrialization, deportation) their proportion is under 25% today.

Komi is a relatively well documented language. The first texts are from the 14th century, and there is a great collection of dialect texts from the 19th and 20th centuries. There are linguistic descriptions of Komi from the 19th century, but hardly anything is described in any of the modern linguistic frameworks.

4.2. Creating a Komi Morphological Description

Since the annotated corpora we want to create are intended for linguists, we decided to use a quasi-phonological transcription of Komi based on Latin script instead of the Cyrillic orthography of the language. The non-phonemic nature of the Cyrillic orthography results in a number of linguistically irrelevant alternations we did not want to deal with in the first place. On the long run,

however, we plan to produce a Cyrillic version of the analyzer as well.

The first piece of description we created in the Komi subproject was a lexicon of suffix morphemes along with a suffix grammar, which describes possible nominal inflectional suffix sequences. One of the most complicated aspect of Komi morphology is the very intricate interaction between nominal case and possessive suffixes.

Another problem we were faced with was that neither of the linguistic descriptions we had access to describes in detail the distribution of certain morphemes or allomorphs. In some of these cases we managed to get some information by producing the forms in question (along with their intended meaning) with the generator and having a native speaker judge them. In other cases we will try to find out the relevant generalizations from the corpus.

Then we started to work on the stem lexicon along with the formal description of stem alternations triggered by an attached suffix. Fortunately, all of the stem alternations are triggered by a simple phonological feature of the following suffix: that it is vowel initial. The alternations themselves are also very simple (there is an *l~v* alternation class and a number of epenthetic classes).

On the other hand, it does not seem to be predictable from the (quotation) form of a stem whether it belongs to any of the alternation classes. This information must therefore be entered into the stem lexicon. The following is a list of all nominal and verbal alternation classes with an example for each from the stem lexicon.

```
töv[N];stemalt:LV;
lym[N];stemalt:Jep;
mös[N];stemalt:Kep;
un[N];stemalt:Mep;
göp[N];stemalt:Tep;
ov[V];stemalt:LV;
lok[V];stemalt:Tep;
jul[V];stemalt:Yep;
```

These are the actual entries representing these stems in the stem lexicon. The quotation form is followed by a label indicating its syntactic category and its unpredictable idiosyncratic properties (in this case the stem alternation class it belongs to). For regular stems only the lexical form and the category label has to be entered.

Irregular suffixed forms and suppletive or unusual allomorphs can be entered into the lexicon by listing them within the entry for the lemma to which they belong. The following example shows the entry representing a noun which has an irregular plural form.

```
pi[N];rr:!Pl;\
     ++!pi+jan[PL];rr:(Cx|Px);
```

The entry defines the noun *pi*, which requires that the morph following it should not be the regular plural suffix (which is *-jas*) and introduces the irregular plural form *pijan*, which in turn must be followed by either a case or a possessive suffix.

In Komi, personal pronouns are inflected for case and reflexive pronouns are inflected for case, number and person. Locative case suffixes can be attached to postpositions and adverbs. Certain parts of these paradigms are identical to that of regular nominal stems, but there are also idiosyncrasies (especially among the forms of reflexive pronouns there are very many idiosyncratic ones). We handled regular subparadigms by introducing lexical features and having the analyzer process the corresponding word forms like any regular suffixed word. Idiosyncratic forms, on the other hand, were listed the in the lexicon along with their analysis.

It turned out to be extremely difficult to acquire any lexical resources (either dictionaries or corpora) for Komi in an electronic form. We found practically nothing on the Internet. At present, we have a very limited amount of text available. We converted this corpus to the quasi-phonemic Latin transcription we use. The stem lexicon now contains all stems occurring in this corpus.

5. Conclusion

The tests we have performed on the corpus available to us with the morphological tools described above promise that they will be an effective means of producing the annotated corpora we intend to arrive at.[2] The morphological database for Komi could be created rapidly using the high-level description language of the development environment. At present, the Komi database contains the description of most of the morphological processes in the language. On the other hand, the size of the stem lexicon is quite limited due to our limited lexical resources.

One of the remaining tasks is to expand and refine the lexicon of the analyzer and to gather further corpora and to annotate them.

References

Beesley, Kenneth R. and Lauri Karttunen. (2003). Finite State Morphology. Stanford, CA: CSLI Publications.

C. Hockett. (1954). Two models of grammatical description. Word 10 (2): 210-234.

J. Hoeksema and R. Janda. (1988). Implications of process-morphology for categorial grammar. In: R. Oehrle et al. (eds.), Categorial Grammars and Natural Language Structures. Dordrecht: Reidel.

P. Matthews. (1991). Morphology. Second edition. Cambridge, MA: Cambridge University Press.

Prószéky, Gábor and Balázs Kis. (1999). A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, (pp. 261–268). College Park, Maryland, USA.

[2] We must also add that beside its fortes, the toolset has its limits: we found that the formalism we successfully used to describe Komi (and a number of other languages) does not apply so smoothly to another small member of the Uralic language family, Nganasan, where a quite morphology-independent surface phonology plays an important role in shaping the form of words. The very productive (and quite intricate) gradation processes in Nganasan are governed by a set of constraints on surface syllable structure (both the presence of a coda and an onset and whether the syllable is odd or even play a role). Gradation in Nganasan is difficult to formalize as a set of allomorph adjacency restrictions because phonemes at the opposite edges of syllables may belong to non-adjacent morphemes. We thus turned to the Xerox finite-state toolset (Beesley and Karttunen, 2003), which fortunately became easily accessible for non-commercial purposes last summer, to create an analyzer for Nganasan.

WISPR: Speech Processing Resources for Welsh and Irish

***Delyth Prys, *Briony Williams, *Bill Hicks, *Dewi Jones, **Ailbhe Ní Chasaide, **Christer Gobl, †Julie Carson-Berndsen, †Fred Cummins, †Máire Ní Chiosáin, ††John McKenna, ††Rónán Scaife, †††Elaine Uí Dhonnchadha**

*Canolfan Bedwyr, University of Wales, Bangor; **Trinity College Dublin, †University College, Dublin, ††Dublin City University, †††Institiúid Teangeolaíochta Éireann

E-mail: d.prys@bangor.ac.uk

ABSTRACT

This paper describes an innovative new project to develop speech processing resources for the Welsh and Irish languages. Funded by the EU's "Interreg" programme, and by the Welsh Language Board, it is a collaboration between the University of Wales, Bangor and Trinity College, Dublin, with input also from University College, Dublin, Dublin City University and Institiúid Teangeolaíochta Éireann, all members of the Irish Speech Group.

Very little work has been done to date on developing speech technology tools for the Welsh and Irish languages. There are no existing usable software packages either for text-to-speech (speech synthesis) or speech-to-text (speech recognition) for either language. Both communities suffer through this lack of resources, and ICT applications for communications and oral information exchange cannot proceed for these languages without the development of speech technology tools.

Developing high quality synthesis and recognition facilities is hampered by the fact that, unlike "major" European languages, there are for Welsh or Irish no corpora of spoken data (high quality audio recordings) annotated at the various linguistic levels (e.g. phonetic, phonological, and grammatical) which is a prerequisite to developing electronic speech communication tools. The provision and analysis of such corpora is the basic building block needed for this task. This project seeks to lay the foundation for future work, taking a modular approach that will re-use language tools to create various speech applications.

Both partners believe that the best way to disseminate speech technology tools in a minority language environment is to provide freely distributable and unlicensed tools that are easy for the end user to adopt. These may also be taken up for further development by businesses to incorporate into other utilities: e.g. educational software, screen readers, or communication announcements on public transport.

The WISPR project concentrates on the development of annotated speech corpora and (in the case of Welsh) providing a text-to-speech synthesis tool as the initial phase of speech technology to be delivered. Text-to-speech (TTS) is cited by both educators and disabled people as being the most urgent tool that needs to be developed for Welsh and Irish. Speech recognition tools and improved TTS tools will be the subjects of future projects. Speech synthesis software will therefore be produced in this project: the basic prerequisites for a first-pass speech synthesis tool in the case of Irish, and a more developed speech synthesis tool for Welsh. The precise targets set for each language reflect both language-specific differences between the two (e.g. the greater complexity of the phonological and orthographic systems of Irish) and the availability of previous groundwork in Welsh that can be exploited in the present project.

1. INTRODUCTION

The WISPR project ("Welsh and Irish Speech Processing Resources") is the first collaboration between Welsh and Irish researchers to develop speech technology tools and resources for Welsh and Irish. The initial focus will be on providing the infrastructure for TTS in both languages. It is funded by the Interreg programme of the EU, together with the Welsh Language Board, and runs until the end of 2005.

The principal partners are the University of Wales, Bangor and Trinity College, Dublin. There is also additional input from Dublin City University, University College, Dublin, and Institiúid Teangeolaíochta Éireann (ITE).

The requirement for tools and infrastructure to build speech technology applications is particularly pressing in the case of minority languages. This is because the languages themselves are threatened to varying degrees. Industry is unlikely to provide the necessary resources, due to the lack of sufficient commercial return. Furthermore, the leverage represented by speech technology is proportionately much greater for minority languages: these very tools have the potential to assist in stemming the decline in language use.

There has been a little work in this area for Welsh in the past. A diphone-based synthesiser has been developed [1], [2], and also a small annotated speech database for Welsh [3], based on read speech recorded in a recording studio.

There are currently no suitably annotated corpora of spoken Irish. It should be noted that there is no standard dialect of Irish, but rather three major dialects which are mutually comprehensible. There is a small digital pronunciation dictionary of Irish, with phonemic representations, which

attempts to provide a standardised form that might be acceptable to different dialects. However, it does not reflect the speech of any particular dialect, and would need considerable adaptation to render it suitable for use in a technological context.

2. LINGUISTIC DIFFERENCES AND SIMILARITIES

The decision to work together was motivated not only by ongoing cultural links, and shared needs and objectives, but also by the fact that the two languages share many linguistic features.

Similarities
Similarities at the grammatical and morphophonemic levels make it advantageous to work together and share procedures and resources.

One such example involves the initial consonant mutations, which are very similar in both languages, but rare among languages in general. Mutation involves a linguistically determined change in the initial consonants of a word. One such example is the nasal mutation of /b/ to /m/. For example, the town "Bangor" can mutate (in Welsh) to "ym Mangor" (*in Bangor*). The equivalent process in Irish is seen in the mutation of "Baile Átha Cliath" to "i mBaile Átha Cliath" (*in Dublin*).

Differences
A striking difference between Welsh and Irish, which has far-reaching implications for any attempt to develop text-to-speech synthesis, lies in the orthographic systems. Whereas Welsh orthography is a fairly good guide to pronunciation, Irish retains many conservative features, which can complicate the mapping between spelling and sound. For example, the word "bhfaighfidh" ("will not get") is pronounced [ωɪ] in northern Irish.

The consonant system of Irish is large and complex. There is a phonological opposition between palatalisation and velarisation, which doubles the size of the sound system compared to Welsh or English. Furthermore, for laterals and nasals in Irish, there are many more distinctions in place of articulation compared to Welsh. Laterals and nasals also have a voicing distinction, which a synthesiser would need to take account of. All these factors mean that any concatenative TTS system will need a very large number of synthesis units for consonants in Irish. This may well affect the type of synthesis technology selected.

3. GOALS AND DELIVERABLES

Short-term goals

The specific short-term technical goals associated with the deliverables of this project are broadly similar for the two languages. However, the goals also diverge in a way that reflects the fact that the languages are at different stages of development in speech research, and also the fact that certain aspects of TTS development will represent a greater challenge to Irish. For example, the very complex and opaque orthography will considerably complicate the formulation of letter-to-sound rules for Irish.

For Welsh, the technical goals are similar to those for Irish, but with more emphasis on developing a working TTS system that can be integrated into application software. Initially, this system will be based on the existing diphone system, but it is hoped to progress to a unit selection synthesis method, in order to improve the quality of synthesis. In addition, it is hoped to widen the variety of voices modelled: i.e. male, female and child voices.

It is also hoped to collect and annotate a more extensive Welsh speech database than was previously possible. This database will then form a foundation for future linguistic and technological research for Welsh.

For Irish, a major goal is the development of an annotated Irish speech corpus, initially focussed on the dialect of Connemara. This corpus will then facilitate the second goal, namely developing the prerequisites for Irish TTS (such as: an initial pronunciation lexicon for the dialect, a set of letter-to-sound rules, speaker recordings, and initial prosody models).

Long-term goals

A long-term overriding project goal will be the building of a research community of speech researchers in the Celtic languages, together with obtaining maximum value from limited resources by actively sharing best practice. A further goal which has long-term implications concerns the establishment of links with potential user groups in these languages (of which more below).

4. SPEECH DATABASE ANNOTATION

The database to be collected for each language will be tuned towards the needs of concatenative synthesis rather than speech recognition. Therefore the recordings will be made in a studio rather than in an office environment, and only a small number of speakers will be recorded, with a large amount of data from each one.

It is also hoped that the databases will supply the raw material for descriptive linguistic work on each language. Therefore a great deal of care will be taken over the labelling of each corpus. Manual segmentation and labelling will be used at the lowest level (the acoustic phonetic level) rather than automatic segmentation. This is because, for purposes of linguistic research, it is imperative that segment boundaries and labels should be accurate, and it may prove easier (though by no means faster) to achieve high accuracy using a manual segmentation procedure.

For linguistic levels above the acoustic phonetic level, it may well be easier to make use of the semi-automatic hierarchical labelling facilities in "Emulabel" [4],[5]. This method was previously used to speed the annotation of a small Welsh speech database [3]. It allows for the annotation of a segmented speech database at any levels

previously specified. In the case of prosodic structure, for example, the TOBI intonational labelling system can be used [6], as in the figure below.

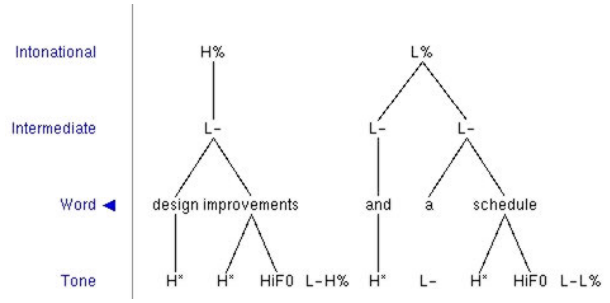


Figure 1: Hierarchical labelling of a phrase using the TOBI prosodic system (from emu.sourceforge.net).

Once the database has been labelled at all levels, it will then be possible to use the EMU Query Tool to extract segments with the desired linguistic characteristics, with reference both to flanking segments, and also to units above and below in the hierarchy. This flexibility allows a great variety of linguistic units to be specified and isolated into a sub-corpus consisting of: unit label, start time, end time. Units could be phonetic segments, or syllables, or words, or even larger units, depending on the particular linguistic phenomenon to be investigated.

Once the units of note have been isolated, it is then possible to carry out statistical investigations, using the raw speech waveform and the F0 trace to provide information on parameters such as: formant structure, F0, loudness, duration, mean energy distribution, etc. In the case of minority languages, basic acoustic descriptions of the speech sounds can be few or non-existent, especially when the different regional varieties are considered. It is hoped to expand on an existing method [3] for rapid prototyping of basic descriptive statistics at the acoustic level. This method would then be available to researchers in other minority languages.

5. LINKS WITH USER GROUPS

As mentioned previously, an integral part of this project will be the establishment of links with probable end users. A primary focus is on the needs of disabled Welsh and Irish speakers, who at present are unable to access computing resources in these languages. In the case of the visually impaired, there is a strong expression of interest from both the Wales Society for the Blind and RNIB Cymru. A similar situation holds in Ireland.

Educational software for language teaching and literacy aids is a particularly targeted area. Whereas in a widely spoken language speech synthesis may be relatively little used in these applications, the potential benefits for a minority language could be far-reaching. For example, Irish is taught as an obligatory subject in the primary and secondary school curriculum, but often the spoken Irish proficiency of the teachers can provide a poor model, far removed from native speaker production. While we are not suggesting that the provision of synthesis will cure all ills in this area, it is envisaged that interactive learning tools, games, etc using high-quality Irish synthesis, could provide major support in the acquisition of more native-like accents. A further area of interest is the application of speech technology for the ordinary user in a home or office environment, using widely available consumer software.

To this end, preliminary links have been established by the Welsh partner with several small companies in Wales, and it is hoped to disseminate the results of this work to a wide variety of potential stakeholders. Similar efforts will also be made by the Irish partners.

6. FUTURE DIRECTIONS

This project is seen as the beginning of what we envisage will be a much larger enterprise encompassing the broader Celtic speech community.

In the first instance we would envisage consolidating the output of this project by adding additional voices for the initial dialects. We also envisage covering other dialects of Irish and Welsh . In Irish we hope to extend coverage to Donegal Irish and Munster Irish, and in Welsh to provide at least a voice each for north and southern varieties..

In the broader picture, it is hoped that the project will build a foundation that can then be extended to the other Celtic languages (Breton, Cornish, Scottish Gaelic). We have already established a Celtic Speech Group, and are in touch with colleagues working with Breton and Scottish Gaelic so that they in turn might reap benefits from our current work and eventually collaborate on future joint initiatives.

Looking even further ahead, the aim is to develop a general procedure for the following tasks:

- Designing and recording a speech database.
- Using the speech database to carry out basic linguistic and statistical research into the language.
- Integrating the TTS system into application software.
- Building a text-to-speech synthesiser.

This procedure would be fully documented, and computer resources (scripts, rules, etc) would be made available, in order to smooth the path for any future researchers in other minority languages. This alone would represent an immense gain for the field of minority languages in general.

ACKNOWLEDGEMENTS

This project is funded by INTERREG, an initiative of the European Union to facilitate co-operation between adjacent regions. Additional funding has been provided by the Welsh Language Board.

REFERENCES

[1] Williams, B. "Diphone synthesis for the Welsh language", *Proceedings of the 1994 International Conference on Spoken Language Processing*, Sept 1994, Yokohama, Japan.

[2] Williams, B. "Text-to-speech synthesis for Welsh and Welsh English," in *Proceedings of the European Conference on Speech Communication and Technology* (Eurospeech 95), Madrid, Spain, 18-21 Sept. 1995.

[3] Williams, B. "A Welsh speech database: preliminary results". *Proceedings of Eurospeech 99*, September 1999, Budapest, Hungary.

[4] The EMU Speech Database System: http://emu.sourceforge.net/

[5] Cassidy, S. & Harrington, J. "Emu: an Enhanced Hierarchical Speech Data Management System". *Proceedings of the Australian Speech Science and Technology Conference,* 1996, Adelaide, Australia.

[6] Beckman, M and Ayers G. Guidelines for ToBI Labelling. www.ling.ohio-state.edu/research/phonetics/E_ToBI/

IMPERIAL AND MINORITY LANGUAGES IN THE FORMER USSR AND IN THE POST-SOVIET AREA

Rajmund Piotrowski, Yuri Romanov
Herzen State Pedagogical University of Russia, St.Petersburg, Russia.
48, Moyka Emb., St.Petersburg, 191186, Russia
E-mail: Yuri Romanov <romanov@YR4993.spb.edu>

ABSTRACT

The history of development of the "language construction" in the Soviet Union and, later, in the post-Soviet areas, yields rich data allowing to observe the process of struggle or balance between the dominating "imperial" language and the minority languages. The cultural and stylistic prestige of the "imperial" language works as a linguistic AIDS virus paralyzing the immunity synergetics of a national minority language. First, the status of the latter is reduced to a level of an every-day speech jargon, and then comes the period of the so called "Irelandization", i.e. the national language goes out of use, and the ethnic group turns to the exclusive use of the "imperial" language. In a few generations, such an ethnos can disappear entirely.

We should take our chance to study the synergetic mechanisms influencing the life and death of a language in our efforts to help the national minority languages to survive.

1. A STRUGGLE WAGED BY THE MINORITY LANGUAGES AGAINST THE DOMINATING ONES

It would be expedient to study the fates of the minority languages, suffering from the pressure of the dominating (official, "imperial", "global") languages in their struggle for survival, in the trend of synergetic ideas. Synergetics is known as an interdisciplinary trend studying general principles, methods and techniques of self-organization, self-development and self-perfection of complex systems of various nature, including social and communicational-linguistic ones (Haken, 1983).

Important data on functioning of linguistic synergetics can be acquired by means of direct observing and analyzing failures in its working mechanisms, which may lead to the extinction of a language. We can distinguish among four different ways of such extinction.

The first way is connected with disappearance of a language as a result of:

- either the complete extermination of the aboriginal native speakers (cf. the fate of the natives of Tasmania and their native language),

- or a short-term drastic suppression of an *aboriginal* language by a prestigious language of colonizers or conquerors (let us call the latter an *imperial* or *dominating* language). Such examples we can see in the fate of the many aboriginal dialects of the Indian tribes in Central and South America in the 16th century, or in disappearance of the German local dialects on the territory of the USSR in the 1940s - 1950s. In such cases, the transience of the process of a native language disappearance makes it practically impossible to follow the details of its ruining or deformation of its synergetic mechanisms.

The second way is marked with conservation of a language in the sphere of Cult or administrative function with its simultaneous going out of use in everyday colloquial practice and in the national literary process. As an example of this, we can mention the use of the Latin language in the Roman Catholic Church ceremonies and, also, in the official, judicial and diplomatic practice in the majority of the European countries in the Middle Ages. In this case, for lack of a live colloquial communication, the development of the language gets frozen, and its speech standards are conservated. In this condition, it is impossible to study its synergetic dynamics.

The third way is a gradual retreat of a minority language in the bilingual situation. It is conditioned by structural-typological or communicative advantages of the dominating language which suppresses the use of the minority language. These very circumstances seem to be the reason why the Turkic languages (Karachai, Balkar, Koumyk), with their clear agglutinative typology, took root in the Caucasus pushing aside, in some North-East areas, the complicated inflexional-polysynthetic speech of the Circassian and Avarian autochthons. Also in this sense, very interesting is the fact that the Akkadian language, whose cuneiform writing was very complicated and difficult to learn, gave way to the Aramaic language using a letter alphabet for writing, which was much easier to learn, since the 9th century B.C. Unfortunately, at present there are no descriptions of structural-diachronic details of such processes as yet.

The fourth way of a language's fading away is characterized by a gradual retreat of aboriginal speech under the cultural or social-administrative pressure of a prestigious imperial language. This retreat begins with atrophy or stagnation of "high" styles (administrative-judicial, scientific-technological, etc.) and ends either with a complete reorganization of the native colloquial speech or with its extinction.

The phenomenon of retreat or, sometimes, dying out of the minority languages is characteristic not only of the former USSR or of the post-Soviet area, but of all continents. On the average in the world, up to twelve minority languages disappear every year (Потапов, 1997: 3 - 15).

The most typical situation with suppression of the minority languages is their ousting by the official governmental language. One can see this in Israel where the governmental Hebrew is ousting Yiddish and Ladino. However, the position of the dominating language can be taken not only by a governmental language or by that of the table nation, but also by some other language possessing high cultural-economic and social prestige. For example, in the republic of Eire, only about 200 thousand people can speak the governmental Irish language, the rest

3.5 million inhabitants prefer to use English as a means of communication (Gunnemark, 1991: 31-32). A paradoxical linguistic situation could be observed in the 19th century in the Western districts of the tsarist Russia formed on the territory of the former Great Princedom of Lithuania. The Polish language here, suppressed by the tsarist administration, was not an imperial one for the Lithuanian and Byelorussian languages. Nevertheless, for them, it was a prestigious dominating language remaining as such up to the beginning of the 20th century and serving as a model for creation of their national orthography and terminology.

It also happens that one minority language dominatingly suppresses another native language. An example of this case presents the linguistic fate of the Chuvan Yukagirs who lived in the area of the Lower Kolyma river in Eastern Siberia. About a hundred years ago, they forgot their native Yukagir language and turned to the use of Chukotic - the native language of the neighbouring minority people. A bit earlier, a similar fate overtook several other minority nations who lived in the Yenisey river area. They underwent the Turkic influence, and out of this the Shor and the Khakass languages developed in the Altai area.

2. LANGUAGE "CONSTRUCTION" IN THE SOVIET UNION AND IN THE POST-SOVIET AREA

The history of development of the "language construction" in the Soviet Union and, later, in the post-Soviet areas, yields rich data allowing to observe the process of struggle between the dominating "imperial" language and the minority languages. The process can be subdivided into three periods.

At first, in the Trotsky-Lenin period (the end of the year of 1917 - the beginning of the 1930s), the Soviet authorities fostered development of the languages of the dependent minority nations. The development included not only their literary standards, but also administrative bureaucratic, science and technology styles supported by sets of corresponding terminology. The second, the so-called Stalin period (1930s-1980s), was marked by an overwhelming orientation to the use of the imperial Russian language as a means of administrative bureaucratic, science and technology, and cultural communication with simultaneous ousting the minority languages from these spheres of people's activity.

At the third, post-Soviet, period (starting from the end of the 1980s), the majority of the minority languages of the former USSR, some of which acquired the status of a corresponding official governmental language, came to a defective, most often - destroyed, style system. Thus, in the Caucasus republics of Georgia and Armenia, in the Turkic republics of Central Asia, in the Western Ukraine, the national languages are mostly used in the spheres of every-day colloquial speech, national belles-lettres and social and political journalism. As for those of science and technology, administrative bureaucracy (especially, the juridical and military aspects) and business, the use of the Russian language prevails. In the regions of the Northern Caucasus, the Ukraine, Karelia and Moldova, the national languages function only as rural and urban low colloquial means of communication. As for the Volga region, Siberia, the Far North and the Far East, out of 120 minority languages existing there, more than a half of their number are on the threshold of extinction (Скворцов, 1995: 20-21; Красная книга..., 1999). It is a dangerous situation as it leads to exhaustion and then to the loss, by the people, of their national culture and their ethnic identification.

3. MECHANISM OF DESTRUCTION OF THE MINORITY LANGUAGES SYNERGETICS

Destruction or hampered development of the science and technology and administrative bureaucratic aspects of the minority languages under the influence of the dominating imperial language, have not only cultural-ethnic, but also intralinguistic structural synergetic consequences. The stylistic prestige of the imperial language works as a linguistic AIDS virus paralyzing the immunity synergetics of a national minority language. First, the status of the latter is reduced to a level of an every-day speech jargon, and then comes the period of the so called "Irelandization", i.e. the national language goes out of use, and the ethnic group turns to the exclusive use of the imperial language (cf. the fate of the ethnic Germans in the Volga region, or that of the population of Byelorussia). Thus, in a few generations, such an ethnos starts agonizing and can disappear entirely. We can see that in the fate of the ethnic Germans in the Volga region, the Moldavians of the Ukraine, Abkhasia and Russia, the Pontic Greeks and the aboriginal population of Byelorussia.

The communicative-linguistic mechanism of the process looks like this. Since a national minority language does not dispose of the administrative bureaucratic and science and technology styles, or those styles are in a rudimentary or fading away state, in the conditions of professional or business communication, the native speaker has to turn to the imperial language possessing well developed means of communication and special terminology, in the first place. So, in many former Soviet republics, to discuss scientific and technological problems or in administrative communication, they use "Pidgin" or Volapük jargons, where the national vocabulary is mixed with non-adapted Russian terms and neologisms.

Here is an example of the Finnish-Russian volapük spoken by the Izhora people living in the Western part of the Leningrad region:

...nüt käüv niin ХАЛТУРИТ näille ДАЧНИКKoille

'...теперь ходит, халтурит у этих дачников' (...now he comes to work for those dacha residents). The Izhorian speech is rendered with the Roman letters, and the Russian insertions are shown with the Cyrillic ones.

Similarly, the Moldavian population of the Dniester river region form their Roman-Slavonic pidgin:

Cf.: *Eu sapam ВСЁ ВРЕМЯ ПРОТИВОТАНКОВОЕ...ПОТОМ, ЗНАЧИТ, noi ЗАРАБАТЫВАЛИ pentru ДОСТАУКа materialulu...Apoi eram СОВЕТСКИЙ СЛУЖАЩИЙ, avem documentu, lucrau pentru ОБСЛУЖИВАТЬ НАСЕЛЕНИЕ...om fost numit ca НАЧАЛЬНИК ПОЧТОВОуО ОТДЕЛЕНИЯ...*

'я копал все время противотанковое…, потом, значит, мы зарабатывали на доставке материала... Затем я был советским служащим, имел документ, работал по

обслуживанию населения,... был назначен начальником почтового отделения...'
Or, here is another instance:
НЕ ПРЕДВИДИТСЯ asta să şii ghini
'не предвидется, чтобы было хорошо'.

Such pidginized hybrid jargons often appear when "crossing" two cognate languages, one of which proves to be dominating. In 2000-2003 E.Eranowska-Gronczewska (2004) studied the Polish speech as spoken by the Poles living in St.Petersburg, Russia. She offers the following example:
Cf.: *Chińczyk ИГРАЛ ДО УПАДУ*
(instead of the Polish: grał do upadłego).

3.1. INUNDATION OF LEXICAL BORROWINGS FROM THE IMPERIAL LANGUAGE AND DESTRUCTION OF THE PHONEMIC STRUCTURE OF THE MINORITY LANGUAGES

Valuable data for disclosure of the synergetic pressure directions of the imperial languages upon those of the minority nations is revealed by the study, conducted by the modern Turkic philologists, of the process of adaptation of the flow of the Russian and, also, international terminological borrowings in the Turkic languages (Исенгалиева, 1966; МАТЕРИАЛЫ, 1995 et al.).

One of the basic features of the Turkic languages system is the use of words consisting mainly of one or two syllables. Whereas the Russian, as well as the international, political, professional and science and technology terms are adopted here as indivisible lexical stems with all their prefixes, suffixes and inflexions.
Cf., Kazakh *сингармовариация, сингармофонология* (МАТЕРИАЛЫ 1995: 52 -53), Uigur *денационализация*, Chuvash *самообслуживани* 'самообслуживание'.

Such Russian and international borrowings in a Turkic context exist as a kind of alien cumbersome indivisible lexemes. In the process of interacting with the Turkic predicatives, they form analytical equivalents of the Russian infinitives.
Cf., Tatar *имитация ишлэу* 'имитировать', *муниципализация ясак* 'муниципализировать',
Chuvash *шлихтовка ту* 'шлихтовать' , etc.
Such lexemes can accept Turkic affixes (at the right end). Confer the nominal word combinations and word forms: Kazakh *психологиялық-педагогикалық терминдер* 'психолого-педагогические термины', *химиялық элементтер* 'химические элементы',
Chuvash *акробатсем* 'акробаты', *космонавтсем* 'космонавты',
or "the infinitives" like: Kazakh *денационализациялау*, Kirghiz *денационализациялоо*,
Turkmen *денационализирлемек* 'денационализировать', etc.

However, the imperial language influence upsetting the minority languages synergetics manifests itself not only in adoption of cumbersome lexemes alien to the Turkic languages, but also in a growing number of cases of violation of synharmonic system which is the basic and most important means for Turkic word-form creation.
Cf., Kazakh *аранжирлеу* 'аранжировать' (with palatal synharmony) goes side by side with *стажирлау* 'стажироваться' (without palatal synharmony);
Chuvash *децентрализациле* 'децентрализация' (with palatal synharmony), but *аргументла* 'аргументировать' (without palatal synharmony).

At present, in some Central-Asiatic republics which appeared on the territory of the former Soviet Union, they are making attempts to renew the national terminology by way of replacement of Russian and Latino-Greek international terms by short lexemes of the Turkic or Arabian-Persian origin. Though it is not always that such initiatives receive a unanimous approval by the national terminologists.

Overflow of the imperial lexical borrowings in the minority languages which are mostly used without a phonetic assimilation (the original imperial phonetic version sounds more prestigious) distorts the native phonetic system and leads to disorder in their phonological system. These violations usually entail a loss of some more important system mechanisms. A vivid evidence of this phenomenon is distortion of synharmonic mechanisms which are the most important means for singling out and marking the boundaries of a Turkic and Finno-Ugric word. Also, as a result of a mighty flow of lexical borrowings from the literary Persian and other Iranian languages into the Turkic languages, the Uzbek language and some of its dialects completely lost their synharmony. The same fate befell the Karaite language which underwent influence first of the Polish and then of the Russian languages.

So, all these data reveal the symptoms of instability in the systems of some minority languages which could mean degradation of their synergetic mechanisms. In the process of interference of the imperial and the minority languages bringing synergetic instability in the latter, the following three aspects of the language take the most active part: stylistics, vocabulary and phonetics. High stylistic prestige of the imperial language, playing the role, as it was said above, of a linguistic AIDS virus, opens a broad gate for intensive flow of political, professional, science and technology terms to penetrate into the colonial language. The new lexical units usually retain their imperial phonetic features which are often incompatible with the phonetic standards of the colonial language. The new habits of word articulation can cause instability of phonological and then grammatical homeostasis of the latter.

4. FROM THE PIDGINIZED FORM OF THE IMPERIAL LANGUAGE TO PIDGIN AND THE CREOLE LANGUAGES

The process of the language pidginization as described above is of a two-sided nature. The native substratum can influence the imperial (dominating) language as well. Two forms of influence can be distinguished here. At the first step, when learning the dominating language at regular school or catching by ear, the aboriginals bring some elements of the native language, mainly phonetic-phonologic and grammatical, into the imperial speech.

Another, and stronger, deforming factor of a native minority language influencing the imperial one is its pidginization at mass interethnic contacts when the

natives, speaking the dominating language , did not learn it at school but acquired the knowledge of it spontaneously by ear. At this stage of pidginization, the most frequently used words and word combinations of the dominating language are mixed with fragments of the native speech. Here are a few lines illustrating the "English" speech of the Soviet (Russian and Ukrainian) émigrés of the 2nd and 3rd generations in the USA (the English word-forms are marked with the capitalized ROMAN):

Отрежь мужчине два SLICEика НАМу; *заSHUTай DOORу, а то CHILDRENята заSIEKуют*.

Or a grandmother tells her grandson: - *Закрой WINDOWKу, внучек. COLD поймаешь!*

A wise "philosopher" of the Ukraine on the sense of life: - *Що ты имаешь в своей COUNTRY? Я имаю CAR, SEVEN CHILDRENят, WIFE*.

Gradually a pidginized version of the dominating language forms a contact language (Lingua Franka) for interethnic communication or, in the modern terms, Pidgin. From the synergetic point of view, Pidgin is a spontaneously and chaotically forming zone of unstable linguistic condition.

Basically, Pidgin uses a reduced vocabulary of the dominating language (lexical stems and initial word forms, to be more exact). The grammar rules are simplified to the limit or even destroyed.

In the situation of a linguistic chaos, Pidgin can either go out of use or, on the contrary, developing its features, it can go in use in the social, ethnic, and thematic spheres of communication advancing, during 3 or 4 generations, into a new dissipative state, and turn into an independent Creole language.

In the Creole languages, in distinction to Pidgin, a chaotic "agrammatical" use of a reduced vocabulary of the dominating language acquires a certain structural organization (Degrees of Restructuring in Creole Language 2001). As an example, we can mention the Kamchadal dialect of the Russian language (actually, a new Creole language) which is spoken by a special ethnic group living on the Kamchatka peninsula whose ancestors were the Russian Cossacks - the conquerors and the native tribe of Itelmen.

5. CONCLUSION

The rapidity of the processes of the language interference, extinction of languages, pidginization and creolezation of the dominating languages make very favorable fields for studying synergetic dynamics.

Unfortunately, the chance was lost to study and describe formation and development of Pidgins and the Creole languages in the period of the 16th - 19th centuries. In the 20th century the interest of linguists to this phenomenon gradually grew up, but we still lack a well elaborated and unified method for organization of its study and description.

So, at present, we do not have enough informational and statistical materials on the dynamics of the minority languages development which could be systematized and on whose basis synergetic-diachronic research could be conducted.

It is very important to work out a reliable, objective and unified method for description of the processes of interference or dying out of the languages, their resistance to the dominating ones. We cannot lose the chance to fix, with the help of that new method, the processes of pidginization and creation of new Creole languages which are still going on in Siberia, on the Pamir and, also, in Central Africa, New Guinea and Polynesia.

REFERENCES

- Degrees of Restructuring in Creole Language/ ed. by I.Neumann-Holzschuh and E. W.Schneider. Amsterdam/ Philadelpphia: John Benjamins Publishing Company, 2001
- Gunnemark E.V. Countries, Peoples and Their Languages. The Geolinguistic Handbook. Gothenburg, Sweden: Länstryckeriet, 1991
- Language Creation and Language Change. Creolization, Diachrony, and Development/ M. DeGraff (ed.). Cambridge Mass.: The MIT Press, 1999
- Ерановска-Грончевска Е. Русско-польская семантическая интерференция (на примере письменной и устной речи поляков и петербуржцев польского происхождения в Санкт-Петербурге). АКД. СПб: РГПУ им. А.И.Герцена, 2004
- Исенгалиева В.А. Тюркские глаголы с основами, заимствованными из русского языка (производные глаголы синтетического и аналитического образования). Алма-Ата: Наука, 1966
- Красная книга народов. М.: Федерация мира и согласия, 1999
- МАТЕРИАЛЫ - сб. «Т¥ркі тілдері терминдерінін**,** компьютерлік қоры». Халықарлық конференциянын, МАҚАЛАЛАРЫ. I - II. МАТЕРИАЛЫ Международной конференции «Компьютерный фонд терминов тюркских языков». Туркістан - Шымкент: Х.А.Яссауи атындаты Халықаралық Қазақ-Т¥рік университеті 1995
- Потапов В.В. К современному состоянию проблемы вымирающих языков в некоторых регионах мира// ВЯ 1997, № 5
- Скворцов Н.Г. Этничность и трансформационные процессы// Этничность. Национальные движения. Социальная практика. СПб.: Петрополис, 1995

ACKNOWLEDGEMENT

The present research is sponsored by the Russian Fund of Fundamental Investigations (RGNF),
project No. 02-04-00195a.

Issues in Porting TTS to Minority Languages

Ksenia Shalonova and Roger Tucker

Outside Echo (Local Language Speech Technology Initiative)
HP Labs, Filton Rd, Stoke Gifford, Bristol BS34 8QZ UK
{ksenia, roger}@outsideecho.com

Abstract

We describe issues that are arising in the Local Language Speech Technology Initiative (LLSTI) where we are porting TTS to languages where commercial organisations are reluctant to take the risk. Currently Hindi, Ibibio, Swahili, Tamil and Zulu are being developed. We propose that the TTS development process can be considered as an optimal start for linguistic documentation of minority languages. Possible solutions for obtaining formalised linguistic knowledge on different levels are discussed.

1. Introduction

1.1. LLSTI Project

There are a number of commercial TTS companies, all of which are steadily expanding the number of languages they offer according to likely markets (Multilingual Text-to-Speech synthesis, 1998). Our current open-source project, the "Local Language Speech Technology Initiative" (LLSTI) is focused on the development of TTS systems (including training program) for those countries, where the market is unproven and economically poor, and there is little hope of a commercial organisation taking the risk.

The goal of LLSTI project is to enable engineers & linguists without any prior experience of TTS to be able to produce good quality, deployable systems, in a reasonable timeframe. The general approach is to provide a set of tools, which are as language-independent as possible, to provide some basic training, and then to guide partners through the development (porting) process.

To be able to carry out this approach successfully, we worked top-down, carrying out the following tasks:

- To understand from the start what TTS problems have to be solved.
- To find out what information is available for each language in reference works and (reliable) publications
- To extract TTS-related knowledge into database

- To identify technological gaps to be filled in
- To develop (semi-)automatic tools to solve the TTS problems in an integrated way
- To investigate possibilities for re-use of modules from existing languages

There is enormous benefit in making all results freely available. This enables a community of interest to be formed, with different people working on different languages and parts of the system, according to their own expertise and interest. LLSTI is committed to enabling and supporting this open-source approach (Tucker and Shalonova, 2004).

1.2 TTS Development as the Way of Linguistic Documentation for Minority Languages

The development of the language-specific modules (grapheme-to-phone converter, morpho-syntactic analyser etc.) in a TTS system are one way of formalising linguistic knowledge, and thus can be considered a form of documentation for minority languages. The modules have the benefit that they can be used as the basis for a range speech and language applications in that language – Text-to-Speech, Machine Translation, ASR and etc. (see Figure 1).

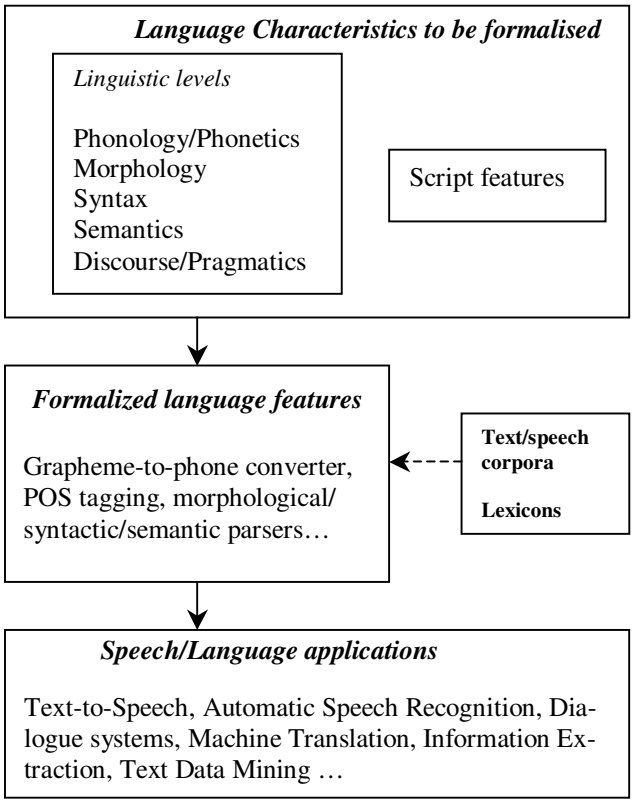


Figure 1: Formalisation of language features for technological transfer between Speech/Language Applications

Formalised linguistic features or language-specific modules can be obtained from annotated text and speech corpora. Data-Driven approaches in Linguistics can be considered as a method both for obtaining new or updating linguistic information.

As the development of TTS systems incorporates formalised knowledge for speech generation on all linguistic levels (phonology/phonetics, morphology, syntax and semantic/pragmatics), it can be considered as an optimal start point for linguistic documentation of poorly investigated languages. As mentioned above, certain linguistic modules and corpora produced for TTS, can be reused in other applications.

1. 3. TTS Development Complexity Score

Performing a survey of languages and scripts used worldwide has enabled us to put what information is available in a database, and identify the problems which will be encountered in building TTS for them. We rank the languages by the TTS development complexity score (Shalonova and Tucker, 2003). This database has formed the foundation of the local language TTS program. The TTS-related complexity for a particular language is calculated by summarising all its script and language feature scores.

Languages / Linguistic Features	Hindi	Ibibio	Tamil	Zulu	Swahili
Tones (Cues for tone assignment)	0	2	0	2	0
Lexical stress (Cues for lexical stress assignment)	0 (1)[1]	0	0	0	0
Secondary stress or rhythm	0	0	0	0	2
Morpho-syntactic characteristics	1	1	1	1	1
Morphological characteristics (derivation)	1	1	1	1	1
Proper syntactic characteristics	0	0	0	0	0
Other characteristics		2[2]		2	

Table 1: Complexity for Language Features

Languages / Script Features	Hindi	Ibibio	Tamil	Zulu	Swahili
Capitalisation	1	0	1	0	0
Consisting Grapheme- to-phoneme rules	1	0	0	1	0
Symbols for loan words	0	0	-0.5	0	0
Symbols for stress	0	0	0	0	0
Symbols for tones[3]	-	0	-	0	-
Punctuation marks	0	0	0	0	0

[1] The existence of Lexical stress in Hindi is disputable.
[2] Terraced tone system related to grammatical characteristics
[3] In combination with the field *Tones* in the table *Languages Features.*

Languages / Script Features	Hindi	Ibibio	Tamil	Zulu	Swahili
Spaces between words	0	0	0	0	0
Homographs	0	1	0	1	1
Other characteristics	0	0	0	0	0

Table 2: Complexity for Script Features

Score / Languages	Intelligibility (basic)	Intelligibility (full)
Ibibio	5	7
Hindi	2 (3)	4 (5)
Swahili	0	5
Tamil	0	2.5
Zulu	6	8

Table 3: Summarized Complexity score

In Table 1 and Table 2 we score Hindi, Ibibio, Swahili, Tamil and Zulu regarding script and language complexity, in Table 3 – we summarise the scores for evaluating the complexity for creation of TTS systems with basic and full intelligibility. By basic intelligibility we mean generally correct grapheme-to-phoneme conversion & stress; full intelligibility also has correct secondary stress, homograph disambiguation etc. The same scale can be applied to all languages worldwide. It should be noted that some linguistic information for particular languages is either missing or contradictory.

2. Characteristics of the developed TTS systems

We are currently developing TTS systems with the following characteristics:

- concatenative TTS
- diphone as a minimal speech unit (as we are using data-driven approach, larger units such as triphones, words and word combinations can be selected from the Speech Database)
- data-driven approach in speech database creation
  - multiple candidates per diphone without prosody modification (currently)
  - combination of a rule-based and data-driven approach – context-sensitive diphones with further prosody modification (in the future)
- uses Festival/Festvox as a basis (http://festvox.org)

In addition to the modules currently available in Festival/Festvox, we provide the following modules:

- Phonetically balanced algorithm for creating an optimal set of sentences for creating Speech Database
- Language-independent Morpho-Syntactic analyser
- Language-independent Intonation extraction/modelling system
- Improved pitch marking tool

- Evaluation procedure tools for testing.

We also support the run-time engine Flite for professional deployment of Festival voices (crucial for telephone applications that need to process several calls at a time and for Windows or PDA-based applications).

The basic training provided currently centers on two major topics:

- speech and text annotation/segmentation
- TTS development overview.

Courses have been held in Bangalore (India) and Bielefeld (Germany). The last course was held with the support of Prof. Dafydd Gibbon.

3. TTS-related problems on different linguistic levels

The current LLSTI partners are developing TTS for the following languages: Hindi, Ibibio, Swahili, Tamil and Zulu. The problem with the languages under development is the lack (or very small amount) of speech and text corpora. It is one of the reasons why their linguistic structure is very poorly investigated, especially such linguistic levels as phonetics (including prosody). Below we present problems on different linguistic levels that we have already experienced while developing TTS systems.

3.1. Phonology/Phonetics

3.1.1. Selecting a normative speaker

TTS systems should normally generate speech that will be accepted by most local people for whom the synthesis is actually developed. For this reason speech databases for TTS are usually recorded by speakers with normative pronunciation. It is not straightforward to define what is normative speech for a language with various dialects (one dialect is normally considered to be the normative pronunciation). The standard pronunciation can be determined by several ways:

- The speech of broadcast readers of central TV/Radio stations can be considered as standard.
- Socio-linguistic study can be carried out. This type of research requires a lot of effort – plenty of recordings and their analysis. Nevertheless, it is the most reliable method as it allows verifying changes in speech culture and thus defining the normative speech (pronunciation standard typically changes significantly over a 20-30 year period).
- "Compulsory" appointment – the speech of a particular person (professor, writer, actor…) can be defined as standard.

For the current project the first method (a broadcast reader/actor of a theatre in a capital city) is taken as a start point as it is the easiest to handle.

For our partners we provide a document for speaker selection procedure that describes the set criteria to be taken into account. It is interesting to notice that for European languages the speaker should normally have a loud and distinctive voice, whereas in Ibibio culture, for example, it is very insulting to speak loudly, so the synthesis will have to replicate a quiet voice with the corresponding voice quality.

3.1.2. Optimal Allophone/Phone inventory and G2P rules for TTS Speech Database

In order to create an optimal speech database for TTS, it is necessary to go through an iterative procedure where segmentation/annotation of the recorded speech material for the speech database itself can change both the phoneme/phone set of a language and grapheme-to-allophone(phone) rules. In this case the data-driven approach for Voice font creation (where the speech database contains a large number of real sentences that have to be segmented/annotated) can be considered as an optimal start point for obtaining new phonetic knowledge about a language.

Ideally, speech corpora required for TTS can be subdivided into 2 parts:

1. Speech corpora for the database itself
2. Speech corpora for phonetic research (including research in prosody).

In our project due to the lack of time we are using only the first type of corpus (approx. 400-600 real sentences) for research purposes and differentiate only between 2 sentence types: declarative sentences and yes/no questions. This strategy is sufficient enough for obtaining preliminary results for segmental (grapheme-phoneme-allophone-phone variation) and suprasegmental (pitch and duration) characteristics.

For each particular language it seems important to find the trade off between the number of allophonic/phone variations (between the number of speech units- diphones) and the degree of detail of acoustic/phonetic transcription required for obtaining natural TTS systems. A great number of phonetic variation in speech due to the influence of nasal consonants, position in a word/phrase/sentence etc. can be represented in the recordings of real sentences for the TTS speech database rather than in the diphone inventory itself. The provided algorithm for choosing Phonetically Balanced Sentences allows taking into account essential phonetic phenomenon while creating the Speech Database.

3.1.3. Implementation of Prosody Rules

Prosody - intonation (pitch variation), duration, stress and syllabification is the most poorly investigated area for the languages under our investigation.

We tried to find an intonation system that is easy to develop from scratch and does not require thorough expertise. Currently we are testing the MOMEL algorithm and INTSINT annotation on several languages (Hirst, (2001). The first results sound promising. We intend to publish them in the near future.

Duration parameters are trained by means of a CART-tree tool (provided by Festival) using the speech database for TTS (400-600 sentences).

It is difficult to define the notion of “lexical stress” for some languages. The difficulty is explained by the fact that acoustically stress is “expressed” by the combination of several parameters such as pitch movement, vowel duration and intensity, which requires thorough phonetic research. For example, there are two contradictory opinions about lexical stress in Hindi - either there is no lexical stress at all or it does exist and depends on the syllable weight. Another problem that is related to lexical

stress is a possible phenomenon of reduced vowels in unstressed syllables. Reduced vowels need to be presented as separate units in the speech database inventory.

3.2. Morpho-syntactic analysis

Currently we are working on the development of a shell for a TTS NLP module. The linguistic specs for this shell are to be filled in by local linguists in India, Kenya, Nigeria and South Africa. The shell is currently a language-independent Morpho-syntactic analyser (details to be published shortly). The analyser has a powerful context-free mechanism that allows to process languages with different morpho-syntactic complexity.

Our experience with Morphological Analyser shows that due to the lack of available lexica for most of our languages, we desperately need a Morphological Learning Tool. This tool has to provide the possibility for obtaining both rules and data (stem and affix dictionaries) on the basis of a limited lexicon (starting from approx. 10.000 units). One of our partners (IIIT Hyderabad) are working on such a tool, but so far it is tested only for Hindi. For Tamil and Swahili a Morphological Analyser is not required for creating a basic TTS system, whereas for Hindi this tool is crucial for prediction of schwa deletion in G2P module.

As for syntactic analysis, in this project we are working on chunking (not full syntactic parser) that will be the basis for assignment of phrases for intonation modeling. We intend to use the same speech database for TTS in order to obtain a preliminary set of rules for phrasing. To the best of our knowledge, phrasing mechanisms for our languages have not been investigated at all. As a start point, we took the algorithms for such commonly investigated languages as English and French (e.g., Black and Taylor, 1994).

Besides phrasing solutions, a syntactic analyser is used in the current project for tone assignment for Ibibio and Zulu. As tones in these languages have grammatical meaning, morpho-syntactic analysis is required.

Linguistic tools related to Semantics and Discourse/Pragmatics are currently not under our development. As the incorporation of such knowledge into TTS will improve Intonation modelling, we hope to deal with this problem in the future.

3.3. Scripts

So far we have not developed TTS for the languages with "complex" scripts such as Arabic with optional vowel marking or Thai with lack of spaces between the words. The only problem that we have experienced is the lack of special symbols for marking tones (basic tones) in Ibibio and Zulu that requires dictionary look-up.

We have experienced an interesting problem with the Ibibio language regarding script and NLP processing. This language does have script, but only a few written texts can be found (mainly several short fairy tails). Prof. Dafydd Gibbon and his team propose creation of written texts/corpora either on the basis of the existing dictionary or by writing down radio broadcasts.

4. Conclusions and future work

In the framework of the LLSTI project we aim to provide solutions for most TTS-related problems that arise on different levels. We are interested in testing our language-independent modules (Morpho-Syntactic Analyser) and techniques (defining of optimal diphone inventory, speaker selection etc.) on a greater number of languages (especially on "lesser investigated" ones).

The lack of linguistic knowledge requires the development of efficient tools (or procedures) for obtaining linguistic rules from scratch. Currently we are testing the Intonation Modelling System for automatic extraction of pitch movements. One of the tasks for the future will be testing different statistical and ML approaches for the TTS modules (e.g., for creating Morphological Learning Tool) and selecting the most appropriate approach on the basis of performance.

5. Acknowledgement

The LLSTI project is currently sponsored by Department for International Development (DfiD), UK and the International Development Research Centre (IDRC) in Canada.

We would like to thank all our partners for their feedback and support: Prof. Dafydd Gibbon – University of Bielefeld (Germany); Prof. Ramakrishnan - IISc Bangalore (India), Kalika Bali – HP Labs (India), Prof. Etienne Barnard, Marelie Davel and Aby Louw – CSIR (South Africa); Prof. Sangal, Dr. Sharma and Dr. Mamidi – IIIT Hyderabad (India); Prof Eno-Abasi Urua and Moses Effiong Ekpenyong–University of Uyo (Nigeria); Dr. Mucemi Gakuru – University of Nairoby (Kenya).

6. References

Black, A. and Taylor, P. (1994). Assigning intonation elements and prosodic phrasing for English speech synthesis from high level linguistic input, ICSLP'94, Yokohama, Japan, pp. 715—718.

Hirst, D.J. (2001). Automatic analysis of prosody for multilingual speech corpora, E.Keller, G.Bailly, J.Terken & M.Huckvale (eds) Improvements in Speech Synthesis, Wiley.

Multilingual Text-to-Speech synthesis. (1998). The Bell Labs approach. Editor R.Sproat. Kluwer Academic Publishers.

Shalonova, K. and Tucker, R. (2003). South Asian Languages in Multilingual TTS-related Database, EACLWorkshop on Computational Linguistics for the Languages of South Asia - Expanding Synergies with Europe, Budapest, pp. 57-63.

Tucker, R. and Shalonova, K. (2004). The Local Language Speech Technology Initiative - Localisation of TTS for Voice Access to Information, Crossing the Digital Divide shaping technologies to meet human needs, SCALLA Conference, Nepal.
http://www.elda.fr/en/proj/scalla/SCALLA2004/tucker.pdf

Creation of a Tagged Corpus for Less-Processed Languages with CLaRK System

**Kiril Simov, Petya Osenova, Alexander Simov,
Krasimira Ivanova, Ilko Grigorov, Hristo Ganev**

BulTreeBank Project
http://www.BulTreeBank.org
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
kivs@bultreebank.org, petya@bultreebank.org, alex@bultreebank.org,
krassy_v@bultreebank.org, ilko@bultreebank.org,ico@bultreebank.org

Abstract

This paper addresses the problem of efficient resources compilation for less-processed languages. It presents a strategy for the creation of a morpho-syntactically tagged corpus with respect to such languages. Due to the fact that human languages are morphologically non-homogenous, we mainly focus on inflecting ones. With certain modifications, the model can be applied to the other types as well. The strategy is described within a certain implementational environment - the CLaRK System. First, the general architecture of the software is described. Then, the usual steps towards the creation of the language resource are outlined. After that, the concrete imlementational properties of the processing steps within CLaRK are discussed: text archive compilation, tokenization, frequency word list creation, morphological lexicon creation, morphological analyzer, semi-automatic disambiguation.

1. Introduction

A corpus annotated with morpho-syntactic information is one of the basic language resources for any language. This is especially true for languages with rich inflectional morphology. The existence of such a corpus is a prerequisite for the development of basic natural processing modules like morphological analyzers, taggers, chunk grammars. Thus, for less-processed languages the compilation of a morpho-syntactically annotated corpus is one of the primary tasks in the area of language resources creation. As a less-processed language we consider a language for which there are no electronic language resources at all or there are some partial ones.

We consider this task as a possibility for the creation of other resources of great importance for natural language processing like morphological lexicons, rule-based disambiguators, morphological guessers, baseline stochastic taggers. In this paper we present a strategy for the creation of a full form lexicon which to be used for morphological analysis of texts in a language. Additionally, the system offers mechanisms for rule- and manually-based disambiguation.

Needless to say, we are far from being pioneers in discussing such a problem. There is a vast amount of literature dealing with the creation of basic electronic resources for different languages: EAGLES initiative, BLaRK initiative. See also (Leech 1991), (Van Halteren 1999) among others. Our aim is not to summarize all the work devoted to this task, but, pointing to the troubleshoots, to contribute with a concrete strategy within a concrete implementational environment.

Languages are very diverse with respect to the complexity of their morphology. According to Bloomfield's classification (how do languages encode information in their morphology?) there are four types of languages: (1) *Isolating languages*; (2) *Agglutinative languages*; (3) *Polysynthetic languages*; and (4) *Inflecting languages*. As mentioned in (Allwood et. al. 2003, p. 4), it is difficult to capture all types of languages within one unified standardized scheme. For that reason, we present a model suitable for the group of inflecting languages. At the same time, this model is re-usable for other groups of languages, with the appropriate modifications. For example, it has been applied to Hungarian and Tibetan along with languages like Bulgarian, French, Croatian.

All the steps are realized in the CLaRK System. In order to facilitate the application of the strategy to a new language we provide a set of examples for English. These examples help the user to learn how to use the CLaRK System and, at the same time, they can be adapted to a new language.

The structure of the paper is as follows: in the next section the architecture of the CLaRK System is presented. In section 3 the general process of morpho-syntactic tagging is described. Section 4 concentrates on the steps of the morpho-syntactically annotated corpus within the CLaRK System. Section 5 outlines the conclusions.

2. CLaRK System

In this section we describe the basic technologies of the CLaRK System[1] — (Simov et. al. 2001). CLaRK is an XML-based software system for corpora development. It incorporates several technologies: *XML technology*; *Unicode*; *Regular Grammars*; and *Constraints over XML Documents*.

XML Technology

The XML technology is at the heart of the CLaRK System. It is implemented as a set of utilities for data structuring, manipulation and management. We have chosen the XML technology because of its popularity, its ease of understanding and its already wide use in description of linguistic information. In addition to the XML language

[1] For the latest version and the documentation of the system see `http://www.bultreebank.org/clark/index.html`.

(XML 2000) processor itself, we have implemented an XPath language (XPath 1999) engine for navigation in documents and an XSLT engine (XSLT 1999) for transformation of XML documents. We started with basic facilities for creation, editing, storing and querying XML documents and developed further this inventory towards a powerful system for processing not only single XML documents but an integrated set of documents and constraints over them. The main goal of this development is to allow the user to add the desirable semantics to the XML documents. The XPath language is used extensively to direct the processing of the document pointing where to apply a certain tool. It is also used to check whether some conditions are present in a set of documents.

Tokenization

The CLaRK System supports a user-defined hierarchy of tokenizers. At the very basic level the user can define a tokenizer in terms of a set of token types. In this basic tokenizer each token type is defined by a set of UNICODE symbols. Above this basic level tokenizers the user can define other tokenizers for which the token types are defined as regular expressions over the tokens of some other tokenizer, the so called parent tokenizer. For each tokenizer an alphabetical order over the token types is defined. This order is used for operations like the comparison between two tokens, sorting and similar.

Regular Grammars

The regular grammars in CLaRK System (Simov, Kouylekov and Simov 2002) work over token and element values generated from the content of an XML document and they incorporate their results back in the document as XML mark-up. The tokens are determined by the corresponding tokenizer. The element values are defined with the help of XPath expressions, which determine the important information for each element. In the grammars, the token and element values are described by token and element descriptions. These descriptions could contain wildcard symbols and variables. The variables are shared among the token descriptions within a regular expression and can be used for the treatment of phenomena like agreement. The grammars are applied in cascaded manner. The evaluation of the regular expressions, which define the rules, can be guided by the user. We allow the following strategies for evaluation: 'longest match', 'shortest match' and several backtracking strategies.

Constraints over XML Documents

The constraints that we have implemented in the CLaRK System (see (Simov, Simov and Kouylekov 2003)) are generally based on the XPath language. We use XPath expressions to determine some data within one or several XML documents and thus we evaluate some predicates over the data. Generally, there are two modes of using a constraint. In the first mode the constraint is used for validity check, similar to the validity check, which is based on a DTD or an XML schema. In the second mode, the constraint is used to support the change of the document to satisfy the constraint. The constraints in the CLaRK System are defined in the following way: `(Selector, Condition, Event, Action)`, where the selector defines to which node(s) in the document the constraint is applicable; the condition defines the state of the document when the constraint is applied. The condition is stated as an XPath expression, which is evaluated with respect to each node, selected by the selector. If the result from the evaluation is improved, then the constraint is applied; the event defines when this constraint is checked for application. Such events can be: selection of a menu item, pressing of key shortcut, an editing command; the action defines the way of the actual constraint application.

Cascaded Processing

The central idea behind the CLaRK System is that every XML document can be seen as a "blackboard" on which different tools write some information, reorder it or delete it. The user can arrange the applications of the different tools to achieve the required processing. This possibility is called **cascaded processing**.

3. Morpho-Syntactic Tagging

Morpho-syntactic tagging means assigning both: a part-of-speech and the bundle of all relevant grammatical features to the tokens in a corpus. Hence, the existence of an appropriate language-specific tagset is needed as well as an initial corpus in the language in question. When considering a less-processed language, several things have to be taken into account with respect to time-frame and financial constraints: (1) re-usability (i.e. the possibility to reuse an already existing resource), (2) representative tagset construction (i.e., when in a group of related languages there exists a tagset for one language, it can be used as a base for the other ones), and (3) using linguistic knowledge background, to minimize the size of the tagset for easier management.

Apart from these prerequisites, several steps have to be performed for the successful morpho-syntactic analysis. They are as follows:

1. tokenization
2. morpho-syntactic tagging
3. morpho-syntactic disambiguation
4. named-entity recognition

These steps require the construction of a set of tools for processing language corpora. The tokenization step requires a hierarchy of tokenizers for handling various cases. The possible combinations should meet some requirements, such as: (1) flexibility with respect to different token types (words, multiwords, punctuation, special symbols); (2) normalization (suppressing the difference between capital and small letters when necessary); and (3) modularity (tokenizing texts of mixed languages).

The morpho-syntactic step can be performed in various ways depending on the language and the existent language resources. The tools can be: taggers, regular grammars and guessers, used separately or in various combinations. A tagger can rely on linguistic knowledge, that is - consulting morphological dictionary. In this case a rule-based guesser is additionally needed to handle the unrecognized words. It usually relies on word-formation principles

and graphical prompts (capitalization, punctuation). On the other hand, a tagger can rely on statistical approaches. In this case it depends on frequency metrics and certain linguistic regularities between words. In state-of-the-art tools, both approaches (knowledge-based and stochastic) are often successfully combined. The regular grammars can be used at least for the following subtasks: (1) tagging multiword expressions (when one morphological word consists of more than one orthographical words), and (2) encoding rich knowledge resources, such as dictionaries.

The morpho-syntactic disambiguation step can be viewed either as a part of the morpho-syntactic tagging, or as a separate module. In the latter case it is performed by a disambiguator, which, similarly to the tagger, can be statistically-based or rule-based. For the creation of a stochastic device, a manually analyzed training corpus is needed. For the construction of a rule-based tool, a preliminary observation over the linguistic phenomena of the language in question is necessary.

Named-entity recognition step can be performed as part of the tokenization level, as part of morpho-syntactic tagging, or as a separate module. At the tokenization level the 'general token classification' can be applied (Osenova and Simov 2002), which distinguishes between common words, names, abbreviations, punctuation, special symbols, errors. Being a part of morpho-syntactic tagging means incorporating gazetteers of names and abbreviations into the tagger. As a separate module named-entity recognition can be organized into grammars and applied over raw or morphologically tagged text.

The order and combinations of the steps, listed above, depend on the language specificities, the aimed granularity of the analysis and on the existent initial resources for the language in question. One possible solution that we propose is generalized and described in the next section.

4. Implementing of Morpho-Syntactic Annotation

In this section we present a strategy for the creation of a morpho-syntactically tagged corpus for a language with little or no language resources. Most of the steps allow for more than one solution, because they depend heavily on the type of the language. Thus, we present a very simple solution which can be a basis for the development of a more sophisticated solution in each concrete case. We give a prompt how the corresponding step can be implemented in the CLaRK System.

4.1. Text Archive Compilation

We consider collecting electronic texts in the language in question as a prerequisite for the creation of a corpus. In order the text to be processed by CLaRK System, it has to be represented in XML format in an encoding appropriate for the language. CLaRK System recognizes Unicode encodings (UTF-8 and UTF-16) and several 8 bits standards for alphabet encodings. It also supports entity converters for several alphabets (ISO 8879).

In the text archive each document has to be marked-up at least to the structural level: chapters, articles, paragraphs. Some additional meta-information would be useful. For this level one can consult: TEI or CES guidelines.

Usually the texts for a given language are available in a plain text, HTML or RTF format. CLaRK System can read plain text or RTF documents directly and converts them into XML documents. The conversion of HTML to XML has to be done outside the CLaRK System, because CLaRK System can read only well-formed XML documents.

4.2. Tokenization

As it was mentioned above, the tokenization is the process of segmentation of the text into sentences and words: (Grefenstette and Tapanainen 1994). In general, the task is quite complex and in CLaRK System we divided it between two tools: tokenizers and regular grammars. The tokenizers work in a cascaded manner. First, a primitive tokenizer is applied which assigns a category to each Unicode symbol, then a sequence of complex tokenizers is applied. Each tokenizer in the sequence works over the output of the previous tokenizer. The tokens for each complex tokenizer are defined via regular expressions. The result of the tokenization is a list of tokens and their categories. This list is an input to the regular grammar tool which actually annotates the text if necessary. At the tokenization level our goal is to segment the text into a list of potential words, punctuation, numerical expressions. We assume that abbreviations, sentences, dates and similar entities are processed at the next level although one can try to do this directly at the tokenization level. Some of the trickier cases like several words forming one token or multi-token words can be processed later.

4.3. Frequency Word List Creation

Having a text archive and a reasonable tokenizer we can select some tokens as a basis for the creation of a morphological lexicon for automatic morphological annotation. This can be done in several ways. We consider the creation of a frequency list of tokens from the electronic texts as a good initial start. Such a word list can be constructed in CLaRK System with the help of Statistical tool. The tool counts the number of occurrences for each token in the selected textual elements. The result is an XML document which contains a table. Each row represents the token itself, its category, the number of occurrences. Additionally, the tokens can be normalized.

4.4. Morphological Lexicon Creation and Morphological Analyzer

Each regular grammar in the CLaRK System has a representation as an XML document. Using XSL transformation it is easy to construct a regular grammar over the word list produced in the previous processing step. Each rule in this grammar searches for a token in the text and substitutes it with an XML fragment. The XML fragment represents the morpho-syntactic annotation. For instance, the rule for the English word 'cost' has the following form:

```
"cost" -> <w aa="NN;VB;VPP;VPT">\w</w>
```

Here the token is on the left side and the XML fragment on the right. The fragment substitutes the token in the text when found. `\w` is a variable for

the input found in the XML document, thus when the rule succeeds the token 'cost' will be substituted with `<w aa="NN;VB;VPP;VPT">cost</w>`. The value of the attribute `aa` encodes all possible morpho-syntactic tags for the given token.

In order to construct such rules for the tokens in the word list the user needs a tagset for the language. The XSL transformation converts the word list into a set of empty rules like:

```
"cost" -> <w aa="">\w</w>
```

The user has to fill the appropriate morpho-syntactic information in them. The help which the system can provide is different sorting over the XML representation of the rules. The sorting can ensure better observation over the tokens. Here especially the reverse sorting (comparing the tokens from right to left) can be useful for grouping the tokens on the basis of their endings. Additionally, one can write conditional insertion operations in CLaRK which to fill the appropriate tags. Such a rule can be *if the token ends in 'lly' then it is an adverb*. Depending on the goal each wordform can be assigned also a lemma. In this way a morphological dictionary for the language is created.

The set of the ready rules is a regular grammar in CLaRK System. It is compiled into a minimized deterministic finite state automaton and can be used for morphological analysis of the texts.

4.5. Semi-automatic Disambiguation

Disambiguation is done with the help of constraints. In the example we use 'some attribute' value constraints. The constraints of this kind determine the value of some attribute on the basis of the context in which the attribute appears. In our case the target attribute is `ana` attribute which represents the morpho-syntactic analysis of the word. The value of the attribute depends on two things: (1) the value of the attribute `aa` for the word in our case which determines all the possible tags, and (2) the analyses of the other words in the text. Thus the first very general constraint states that the value of `ana` attribute is a member of the tokenized value of the `aa` attribute for the same word. This constraint can be used, as it was mentioned above, in two modes: validation mode and insertion mode. When used in insertion mode it will support the manual disambiguation of the annotated text by stopping at each ambiguous word, tokenizing the value of `aa` attribute and offering the user possibility for choosing the right tag. In validation mode the constraint checks whether the values of `ana` attribute is among the tags encoded in the value of `aa` attribute.

Additionally the user can write rules for automatic disambiguation imposing in the constraint more restrictions on the context in which the word appears. For instance, if the word before 'cost' is the determiner 'the', then the value of `ana` attribute for 'cost' is `NN`. Such rules can significantly reduce the amount of human intervention in the process of compiling a morpho-syntactically annotated corpus.

5. Conclusion

In the paper we presented a strategy for the creation of a morpho-syntactically tagged corpus for less-processed languages. The strategy is described within a certain implementational environment — the CLaRK System. The implementation is done as a sequence of steps. All these steps are done in CLaRK System for English. They are described as demos and are part of the distribution of the system. Although the described strategy requires a lot of manual work we think it is a good starting point for the development of more sophisticated approach in CLaRK System. The advantage is that the users have in one place all the necessary machinery for the implementation of each step. It is worth mentioning that the XPath engine of the system also provides an extensive library of mathematical functions which allows the implementation of statistical taggers in the system as well.

6. References

Jens Allwood, Leif Grünqvist and A.P. Hendrikse. 2003. *Developing a tag set and tagger for the African languages of South Africa with special reference to Xhosa.* To be published in the South African Journal of Linguistics and Applied Language.

Gregory Grefenstette and Pasi Tapanainen. 1994. *What is a word, What is a sentence? Problems of Tokenization.* In: Proc. of The 3rd International Conference on Computational Lexicography (COMPLEX'94). Budapest, Hungary. pp 79–87.

Geoffrey Leech. 1991. *The state of the art in corpus linguistics.* In: Aijmer & Altenberg (eds.), English Corpus Linguistics: Studies in honour of Jan Svartvik. pp 8–29.

Petya Osenova and Kiril Simov. 2002. *Learning a token classification from a large corpus. (A case study in abbreviations).* In: *Proc. of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics*, Trento, Italy.

Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, Atanas Kiryakov. 2001. *CLaRK - an XML-based System for Corpora Development.* In: Proc. of the Corpus Linguistics 2001 Conference. pp 558–560.

Kiril Simov, Milen Kouylekov, Alexander Simov. *Cascaded Regular Grammars over XML Documents.* In: Proc. of the 2nd Workshop on NLP and XML (NLPXML-2002), Taipei, Taiwan.

Kiril Simov, Alexander Simov, Milen Kouylekov. *Constraints for Corpora Development and Validation.* In: Proc. of the Corpus Linguistics 2003 Conference, pages: 698-705.

Hans van Halteren (ed.). 1999. *Syntactic Wordclass-Tagging.* Kluwer Academic Publishers.

XML. 2000. *Extensible Markup Language (XML) 1.0 (Second Edition).* W3C Recommendation. http://www.w3.org/TR/REC-xml

XPath. 1999. *XML Path Language (XPath) version 1.0.* W3C Recommendation. http://www.w3.org/TR/xpath

XSLT. 1999. *XSL Transformations (XSLT). version 1.0.* W3C Recommendation. http://www.w3.org/TR/xslt

CLE, an aligned Tri-lingual Ladin-Italian-German Corpus. Corpus Design and Interface

Oliver Streiter, Mathias Stuflesser, Isabella Ties

EURAC, European Academy of Bolzano/Bozen
Language and Law
Viale Druso/Drususallee 1
39100 Bolzano/Bozen, South Tyrol, Italy
{ostreiter;mstuflesser;ities}@eurac.edu

Abstract

Ladin, a Rhaeto- Romance language spoken in the Dolomites, is an official language in South Tyrol and in Trentino. The department "Language and Law" of EURAC has created CLE (Corpus Ladin dl'EURAC), a trilingual corpus with texts in Ladin, German and Italian. It consists of a monitor sub- corpus, intended for terminology research, and of a reference sub- corpus, which will be used for the development of NLP- applications. CLE is stored in a relational database, designed in parallel to the XCES corpus standard. The corpus is accessible via internet through BISTRO, the Juridical Terminology Information System of Bolzano. Queries can be made using regular expressions, and searches can be restricted by further criteria like legal system, passing date, or document type. BISTRO also offers term tools that can be called for the query results: a term recognizer, a term extractor, and a concordancer.

1. Introduction

Ladin is a Rhaeto-Romance language spoken in five valleys of the Dolomites (North-East of Italy). The variants in the five valleys differ with respect to their lexis and spelling conventions. Recently a standardized form of Ladin has been developed (SPELL, 2001; SPELL, 2002).

The recognition of Ladin as official language for administration, legislation and jurisdiction in South Tyrol, and the general advancement of computer-linguistic techniques, have given rise to a number of milestones in the automatic processing of Ladin. Among them are CD-ROM dictionaries of the Ladin varieties Badiot (Mischì, 2001), Gherdëina (Forni, 2003) and Fascian (Istitut cultural Ladin, 2001), internet dictionary interfaces for Standard Ladin[1], Badiot[2] and Fascian[3], the workbench for Ladin lexicography[4], and spelling checkers for Fascian and Standard Ladin[5]. However, important components which are necessary to render a language operational in electronic communication and publications are still lacking.

The department "Language and Law" of EURAC has a longstanding tradition in the elaboration of German and Italian legal terminology, having lead among others, a 4-year project on the standardization of German for its official usage in South Tyrol[6]. For this purpose the so-called CATEX, the Italian-German corpus of national and regional legislation has been created and made accessible trough BISTRO[7].

Among the research partners which promote and facilitate the use of Ladin, EURAC is assuming the task of developing administrative terminology for the variants of Badiot (BA) and Gherdëina (GH). In addition, EURAC aims at providing computational-linguistic software for these two variants. The cornerstone of the activities is the newly created tri-lingual parallel corpus of Ladin, Italian and German (CLE, Corpus Ladin dl'EURAC).

2. Corpus Design, Annotation and Storage

The trilingual corpus is principally made up of official documents. Part of them have been written in municipalities such as orders, regulations and decrees, others are official translations of the Provincial legislation. The Corpus includes also non-legal documents, such as trilingual news reports from the local government and publications provided by the institutes for the development and the conservation of Ladin. Although they do not have the characteristics of legal documents, they are the most authoritative texts in Ladin. The corpus is divided into two sub-corpora: the "Reference Sub-corpus of Modern Ladin" *RC* and the "Monitor Sub- corpus of Modern Ladin" *MC*.

The "Reference Sub corpus of Modern Ladin" includes just authoritative texts, which will be used for the development of NLP-applications. The "Monitor Sub-corpus of Modern Ladin" instead is made up of official trilingual and monolingual documents where the source is not necessarily a Ladin language authority.

[1] http://tales.itc.it:9000/spell/index.html
[2] http://din.micura.it/voc_vb/lad/index.html
[3] http://tales.itc.it:9000/webdilf/index.html
[4] http://tales.itc.it/resources.html
[5] http://www.spell-termles.ladinia.net/ld/download.html
[6] http://www.eurac.edu/About/Projects/2003/index?which=191
[7] http://www.eurac.edu/bistro

The Monitor Sub-corpus will be constantly updated with monolingual and trilingual texts and will be used at EURAC mainly for the descriptive research on terminology. The entire corpus, however, is publicly available in order to promote research on Ladin language. With a trilingual corpus, not only corpus linguistic studies, but also studies in translation science, multilingual drafting and language interference become possible (e.g. Ploner, 2002). Some statistics on the corpus will feature its principal properties (data from 25.03.2004).

The corpus is intended to be balanced according to several criteria. It contains documents from all municipalities of the Ladin valleys. All different kinds of administrative texts such as orders, regulations and records are contained. The subject area treated refers basically to what is the main issue of town halls, i.e. administrative law. In order to create a balance between specialized terminology and common language, as many documents as possible from the Ladin Cultural Institutes are added.

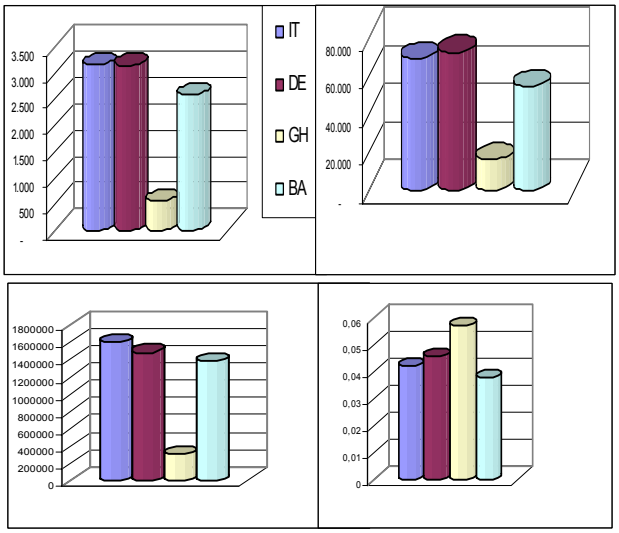


Figure 1: (a) Number of documents per language (b) word types per language (c) word tokens per language (d) type token ratio

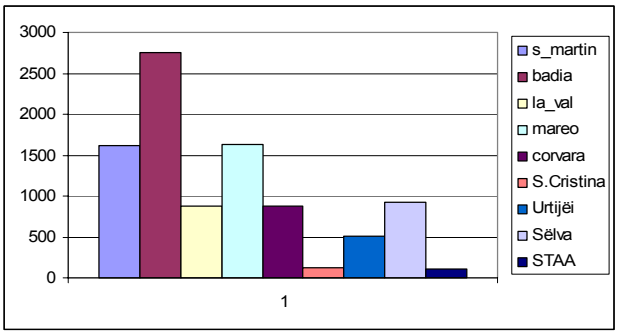


Figure 2: Documents per legal system

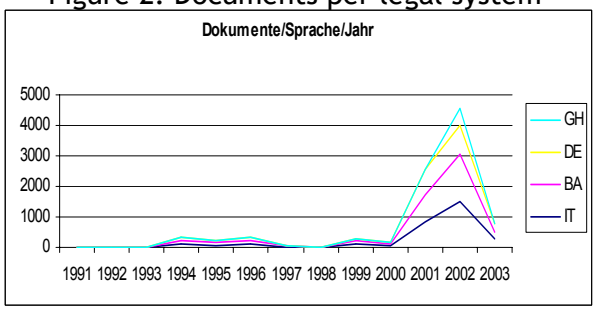


Figure 3: Number of documents per language and publication date

Beside the annotation of the aligned text segments as paragraphs ( ), the corpus is not annotated. The POS-tagging of the reference corpus will be realized in a follow-up project as soon as resources are granted. From the tagged reference corpus a spelling and syntax checker for both the BA and GH variants of Ladin are to be developed. In addition, the Reference Sub-corpus is to be converted and made publicly available in TMX, in order to comply with the local demand for trilingual translations[8]. For this purpose, however, no linguistic annotation is required.

The corpus is stored in a relational database for reasons of maintenance and searchability. Following Bourret (2003) XML is used for data transport only. The set of tuples which are returned as answer to an SQL-query are converted into XML, from which appropriate output structures (XHTML, SVG, PDF) are created (Streiter and Voltmer, 2003). To guarantee a smooth conversion to and from the XML corpus standard XCES[9], the relational database is designed in parallel to this standard. One table corresponds to the document header and describes the meta-data of the document. The meta-data include, e.g., the location of a copy of the original document, in case a validation of the annotated data should be necessary. A second table contains the corpus content, where each cell corresponds to a ' '. A third table contains the alignment. These three tables are unified into a view which can be updated and queried conveniently (cf. Fig. 4). The character encoding of the corpus is Unicode utf-8, as Unicode is the only international standard which covers all diacritic characters used in Ladin spelling.

[8] TMX stands for Translation Memory Exchange. It is an open standard for storing and exchanging translation data between tools with little or no loss of critical data.

[9] The XML-standard XCES contains specifications for the linguistic annotation, alignment and metadata. XCES distinguishes two types of annotation, e.g. a minimal text encoding (sentence, paragraph, heading, etc...) and linguistic annotation (inflection, phrases, etc...).

3. Corpus Search Interface

The interface to the corpus is provided by BISTRO, the Juridical Terminology Information System of Bolzano which, among others, offers the interface to the CATEX corpus and to the database of legal and administrative terminology for Italian, German and Ladin. Tools for corpus-based terminological work such as Internet meta-searches, term extraction and term recognition are equally integrated.

The main purpose of the corpus is to find text segments which contain keywords in one of the three languages. The query is formulated as regular expression. Whenever the query string contains 4 or more consecutive characters of the Roman alphabet, an additional inverted index of character 4-grams speeds-up the search (cf. Damashek, 1995). The first index-search with n-grams approximates the target tuples. The second search with regular expressions refines the search result. N-gram index search doesn't apply to negative, disjunctive or facultative parts of the search term.

Table 1: Operators for the corpus search with regular expressions

	Example	Explanation
Ø	scuola	shortcut for ~* '*scuola*'
~	~ 'scuola'	matches *scuola* , case sensitive
*	~* 'scuola'	matches *scuola*, case insensitive
!	!~ 'scuola'	does not match *scuola*, case sensitive
!	!~* 'scuola'	does not match the string *scuola*,
[ ]	~* 'scuol[ae]'	matches *scuola* and *scuole*
|	~* 'Kinder|Schule'	matches *Kinder* or *Schule*
|	!~* 'Kinder|Schule'	matches neither *Kinder* nore *Schule*
?	~* 'lagh?[io]'	optional h, e.g. *lago* or *laghi*
*	~* 'cun*laburé'	matches zero or more *n'*
+	~* 'col+egament'	matches one or more *'l'*
^	~* '^scuola'	matches *scuola* at the beginning
$	~* 'scuola$'	matches *scuola* at the end

Keywords can be searched in more than one language (e.g. in Italian and in German). This may be useful to disambiguate the searched terms, in the case they have different meanings. The Italian term 'asilo' (kindergarden, asylum, etc), for example, is successfully disambiguated when including the German term 'Kind'. As a result the Ladin equivalent 'scolina' is returned and words like "asil" (en: asylum) are excluded.

The negation marker "!" may be used to find alternative translations, which might otherwise escape our attention among the mass of commonly chosen translations. Figure 5 may serve as an example: Once the standard translation for the Italian 'refezione', the Ladin "refeziun" or "refezion" are excluded, alternative translations show up.

The query may be further restricted by (a) the Ladin variants (BA or GH), (b) the publication date of the document, (c) the legal system, (d) the passing date, (e) the document type and (f) paragraph numbers. A query which involves the publication date will provide diachronic insights. Restrictions on the communities will yield insight into regional variants.

4. Term Tools

All corpus segments, monolingual or trilingual, are accessible via proper URLs. Therefore, the term tools of Bistro, i.e. the term recognizer, term extractor, and concordancer, which operate on URL identified documents, can be called for the query results.

Term recognition refers to the task of identifying the terms and variants of a term base in a text. Term extraction recognizes unknown terms in a text. The example-based method for the term extraction in Bistro is described in Streiter et al. 2003.

5. Corpus Compilation

The original documents have been converted with the help of the open source tools antiword and abiword from DOC-files into two separate TXT.files. Both conversion tools provide different representations which are joined into one "rich" TXT-file. The documents varied from containing perfectly aligned text segments to documents where parallel text segments were disjoint and mixed up with graphics, tables and listings. The extraction of interlingual text segments ("Jahr/Anno/ann 2002" instead of Anno 2002/Jahr 2002/ann 2002) was as common as the toggling of the order of languages ("Jahr/Anno/ann 2002" and "Anno/Jahr/ann 2002") in the same document.

Due to these difficulties, a semi-automatic approach has been followed for the alignment: A program suggests the alignment of 3 document sections on the basis of their position in the document and linguistic triggers. After an interactive confirmation by a linguist, the program stores the decisions and applies them to similar cases. With this approach, regular documents can be processed almost automatically, while irregular documents require the permanent intervention of the linguist. If no model can be applied to the document structure, document sections are aligned manually.

6. Acknowledgements

The work reported herein was carried out mainly within the project 'TermLad II' in the framework of the Interreg III program, funded by the European Union.

7. References

Bourret, R., 2003. *XML and Databases*, www.rpbourret.com/xml/XMLAndDatabases.htm.

Damashek, M., 1995. Gauging Similarity via N-Grams: Language-Independent Sorting, Categorization, and Retrieval of Text. *Science*, 267(5199):843-848.

Forni, M., 2003. *Vocabuler Tudësch-Ladin de Gherdëina, CD-Rom*, San Martin de Tor: Institut Ladin "Micurà de Rü".

Istitut cultural Ladin, 2001. *Dizionèr talian-ladin fascian*, Vich: Istitut Cultural Ladin "majon di fascegn".

Mischì, G., 2001. *Vocabolar todësch - ladin (Val Badia)* San Martin de Tor: Institut Ladin "Micurà de Rü".

Ploner, E., 2002. Ladinisch – Deutsch – Italienische Gesetzestexte. Eine Übersetzungskritik mit Verbesserungsvorschlägen. *Arbeitspapiere der Romanistik* Innsbruck, 13, Leander, M. und G.A. Plangg (Eds.), Innsbruck.

SPELL, 2001. *Gramatica dl Ladin Standard*. Vich/San Martin de Tor/Bulsan.

SPELL, 2002. *Dizionar dl Ladin Standard*. Vich/San Martin de Tor/Bulsan.

Streiter, O., D. Zielinski, I. Ties and L. Voltmer, 2003. Term Extraction for Ladin: An Example-based Approach, *TALN 2003 Workshop on Natural Language Processing of Minority Languages with few computational linguistic resources*, Batz-sur-Mer, June 11-14.

Streiter, O. and L. Voltmer, 2003. A Model for Dynamic Term Presentation, *TIA-2003,* Strasbourg, March 31 and April 1.

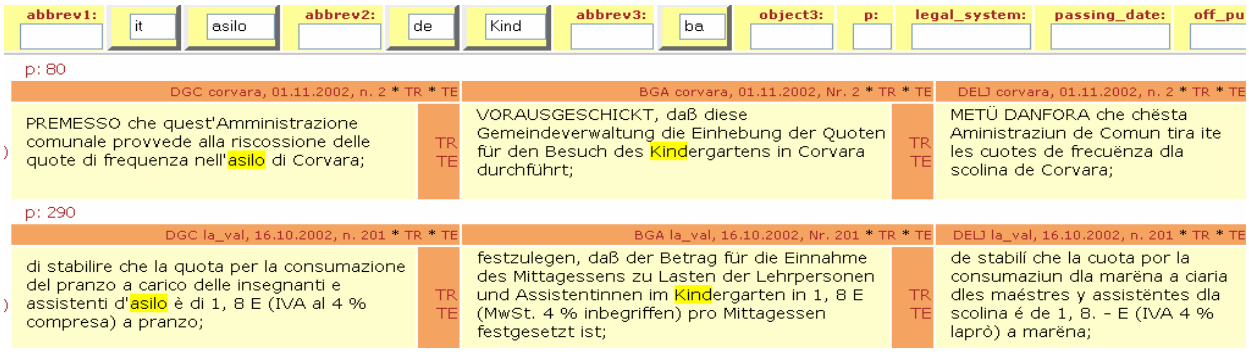


Figure 4: Sense disambiguation with an additional search term.

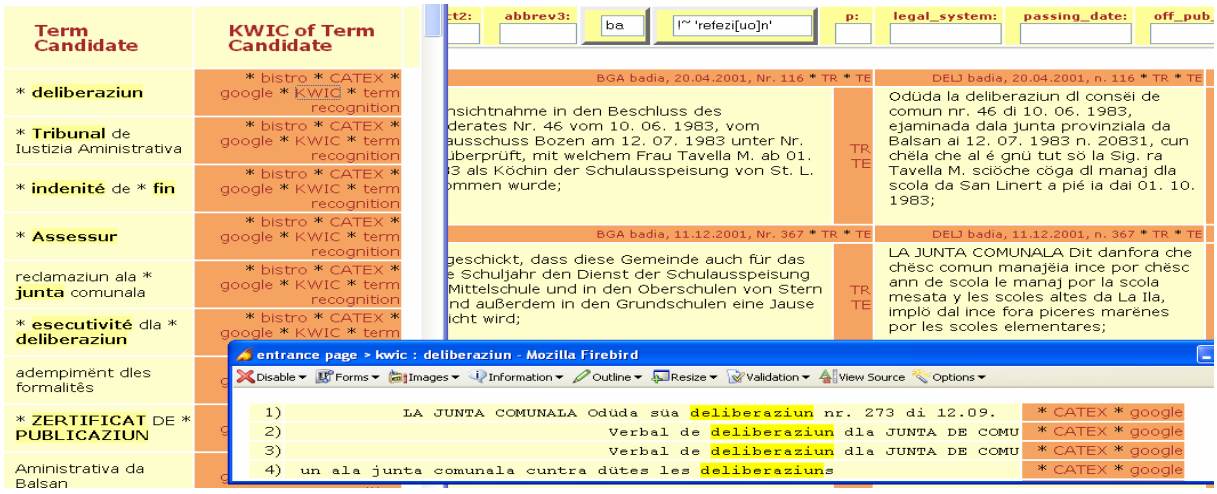


Figure 5: Search of Alternative Translations, Term Extraction and KWIC on mouse click.

Building an interactive corpus of field recordings

Nicholas Thieberger

Pacific And Regional Archive for Digital Sources in Endangered Cultures (PARADISEC)
Department of Linguistics & Applied Linguistics
The University of Melbourne Vic 3010, Australia

thien@unimelb.edu.au

Abstract

There is a growing need for linguists working with small and endangered languages to be able to provide documentation of those languages that will serve two functions, not only the analysis and presentation of examples and texts, but also the means for others to access the material in the future. In this presentation I describe the workflow developed in the course of writing a description of South Efate, an Oceanic language of Vanuatu for a PhD dissertation. This workflow steps through (i) field recording; (ii) digitising or capturing media data as citable objects for archival purposes; (iii) transcribing those objects with time-alignment; (iv) establishing a media corpus indexed by the transcript; (v) instantiating links between text and media using a purpose-built tool (Audiamus); (vi) exporting from *Audiamus* to interlinearise while maintaining timecodes; (vii) extracting citable example sentences for use in a grammatical description; (viii) exporting from *Audiamus* in XML, Quicktime or other formats.

Background

Linguists working on small and endangered languages are being exhorted to produce their data in reusable forms (see for example Bird and Simons 2003) and at the same time to increase the scope of the recorded material so as to document as much as possible of the language and the knowledge encapsulated in the language (see Himmelmann (1998) and Woodbury (2003)). A PhD dissertation focussed on an indigenous language requires a grammatical description that covers the traditionally accepted components of phonology, morphology and syntax. With the use of appropriate tools we can ensure that our normal workload is not significantly increased but that the results conform to those desiderata broadly labeled as language documentation.

In my PhD dissertation, a grammatical description of a language of Vanuatu, I wanted to provide source information for each example sentence and text that would allow the reader to locate the example within the field recordings so as to be able to verify that the example actually did occur in the data. I wanted the fieldtapes (both audio and video) themselves to be accessible To do these relatively simple tasks it was first necessary to link the transcripts of the fieldtapes to their audio source. This audio source needed to be citable with an archival location and a persistent identifier that would endure in the longterm. In 1998 when I began doing this work there were principles and methods pointing in the direction of reusability of linguistic data (as later formulated by Bird and Simons (2003), but the tools to do the work were either nascent or non-existent. By conforming to these principles I have now been able to capitalise on the general linguistic community's development of tools (such as Transcriber, for example). A suitable linking tool has not yet been developed and so I continue to use *Audiamus*[1] (described further below). The thesis was presented together with a DVD of some 3.6 Gb of data, representing over 18 hours of transcribed and linked media data.

Digitisation of the audio file

I conducted fieldwork on the language of South Efate in Central Vanuatu in several fieldtrips from 1996-2000. These fieldtapes contained monologic narratives, conversations and court hearings and were recorded on analogue tape and some digital video with a range of male and female speakers of different ages.

On my return from fieldwork I began digitising my analogue tapes using the built-in soundcard of a desktop computer. This was a mistake! I ended up with digital audio files that I then used to align with the transcript. However, these were not good quality audio files and when the opportunity arose to have the analogue tapes digitised at a higher, and archival, resolution it resulted in me having two versions of the digital data. These two digital versions of the same tape did not correspond in length due both to stretching of the audio tape, and to being played on different cassette players, with slightly different playback speeds. There was no simple correlation between the timecodes in the old version and the corresponding location in the archival version. While I linked all subsequent transcripts to the archival version of the audio file, due to the time constraints of dissertation writing I have kept the non-archival versions for presentation of the thesis data. Archival versions have been lodged with PARADISEC.[2]

The lesson from this experience is that I should have digitised my fieldtapes at the best (archival) resolution possible and then used those files, or a downsampled version, as the basis for linking to transcripts.

[1] On Audiamus, see: http://www.linguistics.unimelb.edu.au/contact/studentsites/thieberger/audiamus.htm

[2] Pacific And Regional Archive for Digital Sources in Endangered Cultures (PARADISEC), http://paradisec.org.au

Linked transcription of the digital media

Audiotapes were transcribed, mainly by a speaker of South Efate, using a ghetto blaster. He wrote the transcripts in exercise books with translations in Bislama, the national language of Vanuatu. These South Efate transcripts were then typed and imported into text/audio alignment software. When I began doing this work in 1998 I used Michel Jacobson's SoundIndex from CNRS/LACITO to align the transcript to the media file (the current tool is Claude Barras' 'Transcriber').

The transcribing software produces a number of different outputs, among them a simple text file which has the utterance chunk together with the start time and end time in the audio file. These linked transcripts are an index of the content of the fieldtape and can be archived together with the media file. However, for the purposes of analysis of the data the links need to be instantiated, that is, we need to be able to click on a sentence and hear it. Using the transcription tool it is possible to do this for each individual file. However, I needed a corpus of all transcripts, with a concordance showing all word forms. There was no such tool available at the time so I used HyperCard to construct the links in a way that allowed the data to be imported and exported easily, using Quicktime to instantiate the links to points within large data files. This tool is called *Audiamus*. Using HyperCard may appear to be a retrograde step, but it capitalised on my existing knowledge of the software, and also allowed me to use a well-developed concordancing tool written by Mark Zimmerman, called 'Free Text'. Combining these tools resulted in a keyword-in-context concordance of texts that played the audio of the context of the selected items (typically the context sentences). The HyperCard tool developed over several years to provide access to the media for the purposes of linguistic analysis.

Audiamus

Audiamus is designed with the key principles of reusability of and accessibility to the data, with the basic premise that every example quoted in the thesis should be provenanced to an archival source if possible. A sample workflow for using *Audiamus* is outlined below, showing that the input is a linked text file and a digital media file and the outputs can be in several textual formats.

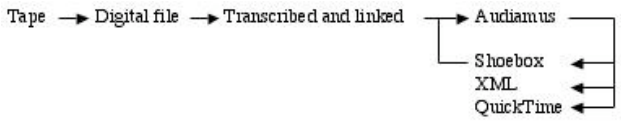


Figure 1: Audiamus workflow

Audiamus allows the user to select an example and to clip either its time codes or both the text and the time codes to the clipboard for pasting into a document. The timecodes are specified as follows: (audio filename, start time, end time) or (98002b,1413.9999, 1419.3600).

Examples can also be clipped with timecodes in a format suitable for processing in Shoebox, e.g.:

\aud 98002b
\as 1413.9999
\ae 1419.3600

The time-aligned text is then available for interlinearising in Shoebox while maintaining the time-codes.

Another export routine provides Quicktime text tracks with time codes in the appropriate format (hr:min:sec:frame) which can be used as subtitles to Quicktime movies.

Examples can be added to a playlist, for use in a presentation for example. The playlist itself can be stored for future use while another playlist is constructed. In the presentation of my dissertation, each chapter was a playlist consisting of numbered examples.

In 2002 *Audiamus* was rewritten (as version 2) in Runtime Revolution, a cross-platform application that builds standalone distribution versions of *Audiamus*. Version 2 can identify media files by their type and by their samplerate, unlike Version 1 in which these needed to be specified for each media file. Version 2 can also play linear mp3 files derived from archival master copies as the same timecodes apply to both.

Conclusion

By adopting the principle of data reusability, I have been able to produce an archivable, citable, extensible set of data and to construct links between my field recordings and the grammar.

For some years now I have observed practitioners using computers for language work of various kinds and, in general, we use what we are most familiar with and what is easiest for us to incorporate into our normal work flow. While this need not be 'bad practice', the crucial point is that the underlying principles of reusability and interoperability of the data are observed regardless of what tools are employed. In the absence of a concerted training effort we must avoid purist dictates about what is THE correct way to proceed and to encourage appropriate use of existing tools until we have purpose-built tools that are generally used and accessible to ordinary working linguists.

If we build data linkage into our workflow as part of normal linguistic analysis, we end up with richer descriptions based on contextualised and verifiable data which has more archival use than does a set of media files or cassette tapes alone.

References

Bird, Steven and Gary Simons (2003). Seven Dimensions of Portability for Language Documentation and Description. *Language* 79: 557-82

Himmelmann, Nikolaus P. (1998). Documentary and descriptive linguistics. *Linguistics* 36: 161-195

Woodbury, Anthony. (2003). Defining Documentary Linguistics (plenary address given at the Annual meeting of the LSA on January 3, 2003) (Ms)

Porting morphological analysis and disambiguation to new languages

Trond Trosterud

Det humanistiske fakultet
N-9037 Universitetet i Tromsø
Norway
trond.trosterud@hum.uit.no

Abstract

The paper presents a parser and disambiguator for North Sami, and effort aimed at porting the work on Sami to other Uralic languages.

1. Introduction

In the past, portability within grammar-based morphological analysis and disambiguation has been hampered by commercial interests linked to good systems. The literature on finite state transducer based analyses typically only describes the mathematical properties of the model, or if more concrete, they only provide toy examples.

Even if the source code of commercial systems used for majority languages were available, they would not have all the characteristics required by new projects aimed at analysing minority languages. Often, they have a long history, thereby containing code written for older systems and they tend to be very large, containing information and resources inappropriate to new parser projects starting from scratch. New parser projects for languages not analysed so far will typically start with contemporary (open-source) tools, and they will start out analysing the core parts of the grammar.

This paper will present a morphological parser and disambiguator for North Sami, a Uralic language spoken in the Northern parts of Norway, Sweden and Finland (the project's home page is *http://giellatekno.uit.no/*. The parser uses Xerox tools (*www.fsmbook.com*) for morphological analysis, and constraint grammar for disambiguation (*sourceforge.net/projects/vislcg/*). The paper will also report from the experiences with porting the system for North Sami to other Uralic languages.

2. The Tromsø disambiguation project

2.1. Morphological analysis

The morphological analyses of the project are based upon a two-level analysis with finite state automata, cf. (Koskenniemi, 1983). We use Xerox software, (*lexc* for lexical analysis and segmental morphology, *twolc* for morphophonological processes, *perl* for preprocessing and *xfst* for case conversion and integrating the parts into a whole (cf. e.g. (Beesley and Karttunen, 1954) and *http://www.fsmbook.com*). The lexical analysis and the segmental morphology operate on two levels, one surface level for roots and affixes, and one underlying level for lexemes and grammatical properties. The surface level again becomes the underlying level for the morphophonological rule set, taking a root- and suffix string as input, enriched with morphophonological information, and transforms it to the wordform we know from the written language (cf. (Karttunen, 1994)).

2.2. Suffixes

The lexicon contains all the roots of the language. The roots are classified according to part of speech (POS) and stem class, directing words taking the same suffixes and undergoing the same morphophonological processes to the same continuation lexica.

2.3. Morphophonology

The dominating morphophonological process in Sami is consonant gradation. Originally, this was a phonological process, where onset consonants in open syllables alternated with a weaker version when the syllable was closed by a syllable-final consonant. Today, the process is fully morphologised. Thus, there is no structural distinction between the suffixes for essive (*-n*) and locative (*-s*) that are added to a noun such as *viessu*, 'house', still, the locative demands consonant gradation, whereas the essive does not. The correct forms are *viessun* and *viesus*. In order to generate this we equip the locative suffix with a diacritical mark indicating weak grade, (`WG`), and make a two-level rule stating that syllable-initial *s* (and some other consonants) are deleted (Cx:0) in contexts where it occurs between an identical consonant and the diacritical mark `WG` (V = vowel, C = consonant):

```
"Gradation: Double Consonant"
 Cx:0 <=> V: _ Cy V (C:) (:C) WG:;
  where Cx in (d1 f l m n n1 r s s1 t1 v)
        Cy in (d1 f l m n n1 r s s1 t1 v)
        matched ;
```

The rule set for North Sami contains 33 rules, and it is ordered according to consonant gradation type, each rule generalising over a smaller or larger set of alternating consonants (in the rule above, 11 consonants may get the value Cx, and thus be deleted.

In practice, this way of doing it led to a time costy compilation process. Lule Sami differs from its closest relative North Sami in having a "Polish", rather than a "Czech" orthography. Where North Sami aims at one letter per phoneme, Lule Sami uses digraphs. When North Sami writes *š* (written *s1* in the rule above), Lule Sami writes *sj*. Since the two-level rules generate over pairs of letters, not over phonemes, Lule Sami get far more consonant alternations than North Sami. The Lule Sami rule set was thus rewritten, from generalising over alternating consonant, to generalising over alternating context for the same consonant. The

Lule Sami rule taking care of the corresponding alternation (*ss:s*) thus looks as follows:

```
''Consonant gradation s:0''
s:0 <=> V: s _ ([j|k|m|n|t]) V: (C:) WG:;
    V: s _ j [k|m|v] V: (C:) WG:;
    V: [b|j|l|m|n|r|v] _ s V: (C:) WG:;
    V: [r|l|m] _ s j V: (C:) WG:;
    V: [r|l|n] _ s k V: (C:) WG:;
    V: [j|m|n|v] _ s t V: (C:) WG:;
    V: r s _ j t V: (C:) WG:;
    V: s: _ #:;
    V: _ Q1: X1: n ;
```

The rule deletes *s* in 9 different contexts. The context corresponding to the north Sami context above is ound in the first line: *s* is deleted between *s* and a vowel followed by the weak gradie diacritical mark.

During compilation, the compiler must resolve a number of conflicts. the number of cuch conflicts is much smaller when we generalise over context rather than over alternating letter, and the compilation process for Lule Sami was much faster than for North Sami (ca. 2 sec. against ca. 2 min 15 sec. on a 640 MHz processor), even though the North Sami rule set contains less consonant gradation rules (33 against 59 for Lule Sami).

3. Disambiguating Sami

Parallel with the work on morphological analysis we started work on disambiguation in the autumn 2003. Due to a recent phonological change (word-final *p, k* were reduced to *t*) there is more homonymy in North Sami than in the other Sami languages (cf. (Trosterud, 2001) for an overview). In languages like Norwegian or English, homonymy is often found across POS boundaries, so that some word-form may be a verb or a noun, but if you know which one it is, you also know the placement within its respective paradigm (we know that *walks* is plural if it is a noun, and that Norw. *landa* is definite plural if it is a noun and not a verb). North Sami homonymy is different. Here, derivation is not done via conversion, but via suffixation, and the homonymy is found within the same POS, and only marginally across POS borders. Whereas disambiguation in English and Norwegian thus may be seen as the task of finding some starting point ("if I am a verb then you must be the noun") the Sami homonymy is grammatical, and more dependent upon neighbouring morphosyntactic properties than upon neighbouring POS disambiguation. 'to throw' is *bálkestit* and 'a throw' is *bálkesteapmi*. The former must be a verb, and the latter must be a noun, but in addition to being an infinitive *bálkestit* may also represent first and third persun plural, and past tense second person singular. Distinguishing verb forms from each other differs from distingushing verbs from nouns, since the contextual differences are smaller in the former case.

As an example, let us take the clause *Mii eat leat dan muitalan* 'We haven't told it', with the verbs *leat* 'to be' and *muitalit* 'to tell'. The sentence is given the following analysis, prior to disambiguation:

```
''<Mii>''
        ''mun'' Pron Pers Pl1 Nom
        ''mii'' Pron Interr Sg Nom
''<eat>''
        ''ii'' V Neg Ind Pl1
''<leat>''
        ''leat'' V Ind Prs Pl1
        ''leat'' V Ind Prs Pl3
        ''leat'' V Ind Prs Sg2
        ''leat'' V Inf
        ''leat'' V Ind Prs ConNeg
''<dan>''
        ''dat'' Pron Dem Sg Acc
        ''dat'' Pron Dem Sg Gen
''<muitalan>''
        ''muitalit'' V PrfPrc
        ''muitalit'' V Act
        ''muitalit'' V Ind Prs Sg1
''<.>''
```

The only unambiguous word is *eat*, first person plural of the negation verb. In reality the sentence does not have 60 readings (2 x 5 x 2 x 3), but one:

```
''<Mii>''
        ''mun'' Pron Pers Pl1 Nom
''<eat>''
        ''ii'' V Neg Ind Pl1
''<leat>''
        ''leat'' V Ind Prs ConNeg
''<dan>''
        ''dat'' Pron Dem Sg Acc
''<muitalan>''
        ''muitalit'' V PrfPrc
''<.>''
```

Here are the rules that were used to arrive at the correct reading (the rules are given according to constraint grammar conventions, the numbers identify positions in the clause, 0 is the wordform to be disambiguated, 1 is the first word to the right and -2 the word two posistions to the left, *-2 to a word two or more positions to the left (for an introduction to the rule formalism, see (Tapanainen, 1996)).

```
SELECT Pers IF (0 ("mii"))
      (*1 V-PL1 BARRIER NON-ADV);
SELECT ConNeg IF
      (*-1 Neg BARRIER VFIN);
SELECT Acc IF (*-1 LEAT-FIN-NON-IMP
      BARRIER NON-PRE-N) (1 PrfPrc);
SELECT PrfPrc IF (*-1 Neg
      BARRIER CONTRA);
```

The pronoun *mii* may be a personal or interrogative. The rule states that if there is a PL1 verb to the left, with no other words than adverbs between the two, then the personal pronoun reading is selected. In order to get the correct reading for copula, the ConNeg form (the form connected to negative verbs) is chosen if a negation verb may be found somewhere to the left, before we find any other finite verb. The rule for perfect participles is similar, but here the barrier is a a set of words cancelling negation, like the word

muhto 'but'. This set has been listed earlier, and is labelled CONTRA. The rule for accusative demands a finite copula to the left, and with nothing but NP-internal pre-modifiers intervening, and a perfect participle to the right. In order to disambiguate running text, approximately 1500 to 2500 such rules are needed.

4. Extending the work on Sami to other languages

4.1. Sharing infrastructure

The main advantage of extending one's work to other languages is of course the benefit of just copying the infrastructure and the makefiles to the next language. We have tested out our solutions on 8 different Uralic languages, in each case reusing the makefiles and the directory structure. File naming procedures are the same for each language, as are the sets of shared grammatical tags. For each new language we have cut the production time by several months, compared to the previous languages. Also, fully or partly language-independent resources, such as preprocessors, tokenizers, databases for names, unassimilated loan words and abbreviations, may be reused.

Porting existing hand-made parsers to new, but grammatically similar languages, offers advantages also when crafting the transducers for the morphophonological and morphological core processes of the languages. Solutions for the structure of continuation lexica, and for similar morphophonological rules, may be reused. And there are many groups of such similar languages. Seen in a diachronic perspective, it takes more than a millennium to create far-reaching structural differences between languages. This means that we should expect to fine closely related languages as a result of large migration and diffusion waves during the last millennium and a half. Such language families are e.g. the Romance, the Turkic and the Bantu languages. Especially in the two latter cases, where several of the languages do not have the same access to written corpora as do many European languages, there are large benefits in working in parallel on several languages.

4.2. Building transducers as opposed to using stochastically-based approaches

Grammar-based disambiguation has been known to provide good results, compared to stochastically-based approaches (Samuelsson and Voutilainen, 1997).

Looking at minority languages, the arguments in favour of grammar-based approaches are even stronger. In the cases of the Sami languages or the Uralic languages of Russia, there is not a choice between using the multimillion electronically available corpus or not. There is no such corpus. Rather, what is available is a grammar, and in most cases a reasonably good dictionary. With these two tools (especially if the dictionary is electronically available, it is possible to build good transducers and disambiguators within a couple of years, or, after a while, within even shorter time. For inflectional languages with hundreds of inflected forms for each lexeme (and sometimes more), transducers based on stem classes and inflectional paradigms are the only way of ensuring good coverage of the language.

4.3. Working on similar languages

One of the main arguments for statistically-based parsing methods has been that it is labour-saving, i.e. that the computer draws the correct conclusion from previously tagged texts (or for certain applications, even from running text only), rather than having the linguists writing the rules by hand. As we have seen, this is not an option for most languages of the world.

With the latest version of the Xerox tools, it is now possible to use UTF-8 in the source code, and thus have access to the full range of Unicode characters. Within the Sami language technology project, we have started experimenting with parsers for languages written with the Cyrillic alphabet (Komi, and partly also Udmurt). Working with UTF-8 on UNIX platforms is still not unproblematic, but the experiences with the core transducers are good.

Another option than the manual writing of transducers is to apply to a combined version of human elicitation and machine learning, as argued by (Oflazer et al., 2000). This approach should be more suited to families of very similar languages, like the Turkic or Bantu languages. Whether these semiautomatic transducers are as easy to update as hand-made ones, or whether they will look more like a "black box", remains to see.

5. Summary

The present article has given an an overview of work with morphological transducers and disambiguators for some related Uralic languages. The work conducted so far shows that the building of transducers and disambiguators will benefit from sharing code written in an as language-independent way as possible.

6. References

Beesley, Kenneth R. and Lauri Karttunen, 1954. *Finite State Morphology*. Studies in Computational Linguistics. Stanford, California: CSLI Publications.

Karttunen, Lauri, 1994. Constructing lexical transducers. In *15th International Conference on Computational Linguistics (COLING-94)*. Kyoto, Japan.

Koskenniemi, Kimmo, 1983. *Two-level Morphology: A General Computational Model for Word-form Production and Generation*. Publications of the Department of General Linguistics, University of Helsinki. Helsinki: University of Helsinki.

Oflazer, Kemal, Sergei Nirenburg, and Marjorie McShane, 2000. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computer Engineering Technical Report*, BU-CE-0003.

Samuelsson, Christer and Atro Voutilainen, 1997. Comparing a linguistic and a stochastic tagger. In *35th Annual Meeting of the Association for Computational Linguistics*.

Tapanainen, Pasi, 1996. *The Constraint Grammar Parser CG-2*, volume 27 of *Publications of the Department of General Linguistics, University of Helsinki*. Helsinki: University of Helsinki.

Trosterud, Trond, 2001. Morfologiija rolla sámi giellateknologiijas. *Sámi diedalaš áigečála*, 3:100–123.