Grammatically based language technology for minority languages

Trond Trosterud

June 19, 2005

Contents

1 Introduction

Languages may live on without a written language. But written languages will not be able to function as administrative languages in modern societies without a developed language technology.

When it comes to computers and language technology, India differs from other countries in significant ways. Although a former colony, it has a long literary tradition, and a civil administration with traditions back to the time before colonialism.

Despite a long literary tradition, very few of the 400 languages of India have written standards. The few languages possessing written languages are poorly represented in localised versions of computer programs, despite India's indisputable strength as an IT nation. Thus, the latest version of Microsoft XP is translated to 33 languages, the smallest of them, Estonian, is spoken by less than one million. None of the languages of India are among the ones listed as languages with a translated user interface. When turning to languages where keyboards and date-time format are included, we find 5 Indian languages, Hindi, Konkani, Marathi, Sanskrit and Tamil. Macintosh has a slightly smaller set of scripts, here we find Devanagari (for Sanskrit, Hindi, Majarathi), Gujarati (for Punjabi), Gurmukhi (for Gurmukhi)

Both compared to smaller European countries and to large Asian countries, the computer localisation of Indian languages thus lag behind. The situation is more parallel to the one of Swahili than to the one of Japanese, Chinese or Korean. This is all the more remarkable, as India is well-known for being a leading country when it comes to computer technology.

The present article will look at ways of building linguistically-based language technology applications for minority languages. It will look at what benefits there are in such applications. Finally, it will discuss the limitations of technology: What is it that languages cannot get via computerization and the internet.

2 Moments

- We need ways of ensuring that when languages are documented, this is done in a comprehensive way
- Written languages need tools based on language technology to function in the computer age
- Grammaticaly based language technology may contribute to achieving both these goals

The last decade has seen a growing interest in linguistic human rights. The background for this is the awareness that speakers of different languages do not have the same linguistic rights. For speakers of different dialects or sociolects of the same language, some may be told that their dialect is "wrong" and in need of "correction". For speakers of different languages, some will find

that they cannot attend school and learn to read or write in their mother tongue, or if they do, they will have to switch to another language when attending secondary school.

3 Transducers as language documentation

Idea:

Building a transducer for generating word-forms gives as a side effect a comprehensive treatment of the morphology of the language in question. A paper-based reference grammar may overlook details on the morphological system, this is immediately revealed in a generating transducer.

At the same time, a morphological automaton may function as input to other language technology applications.

[?] claims that a generative grammar is the same as an explicit grammar. The grammars he had in mind were quite different from the ones advocated here, but the point remains.

All reference grammars run the risk of not being comprehensive.

automant...

The vast majority of the languages of the world are poorly documented.

3.1 What is an transducer, and how can it be built?

A morphological automaton is a d-graph (??), representing all possible word forms in a language.

3.2 Building transducers for languages with Indic scripts

Making such transducers.

4 Language technology - grammatical or statistical?

Moments wrt. choice of language technology:

- Let the commercially interesting languages do the expensive mistakes, and the rest of us may then go for the right solution
- Statistical approaches are good for a language with huge amounts of text electronically available, and a minimal amount of morphology
- Languages with less available texts and more morphology should choose linguistically based methods

5 Language tgechnology preprequisites

Wishlist for a language L that do have a written tradition

1. The letters of the alphabet available in Unicode
2. Localisation resources in place: Keyboard standard, sorting standard, standards for date and time expressions
3. Main software translated into L (word processor, web browser, etc.)
4. Basic language technology applications
   (a) Lexicon
   (b) Morphological parser

(c) Morphological disambiguator

(d) Syntactic parser

(e) Word-sense disambiguator

5. Applied language technology tools for L

   (a) Spell checker

   (b) ...

5.1 Linguistically vs. statistically based language technology

The goal is to give the computer access to as much of our knowledge of our languages as possible. This implies the lexicon, the rules for combining roots and affixes into word forms, and the rules for combining the word forms into sentences.

I will here argue for a linguistically based approach to parsing. Reasons ...

The grammar may be represented as a set of finite-state automata, going from an initial state, via prefixes, roots and suffixes, and finally to the final state (the end of the word-form). Morphophonological processes, such as Umlaut, Sandhi phenomena, etc., may be handled in a separate component. The result is an automaton that gives all the

The tools for such analyses are available from different sources. One is [**?**]. Another is [**?**].

The result is an automaton that may both generate and analyse. It may tell us that (examples). It may tell us that Tamil *example* is a noun or a verb, but in order to find out which one, it needs context.

One way of disambiguating morphological homonymy in running text is via *constraint grammar*, a framework initiated by Fred Karlsson ([**?**] [**?**]), and further developed by e.g. Pasi Tapanainen ([**?**])

5.2 Case study: The Sámi languages

1. Telling the story of the Sámi languages.

2. Then drawing parallels to South Asia

(perhaps written as a case study with the Sámi languages):
(I will attempt at making the presentation concrete, as a cookbook.)

6 Applications

7 Limitations

1. Why language technology cannot save a language from dying

2. How language technology can help