

Using existing written language analyzers in understanding natural spoken Finnish

Tommi Jauhiainen

Department of General Linguistics
University of Helsinki

Abstract

In this paper we consider the possible use of existing linguistic (mainly morphological) analyzers for written Finnish in order to create a system that uses speech as its interface. We also present means to enhance the usability of these analyzers in this respect.

1 USIX Interact–project

In the USIX Interact project ("<http://www.mlab.uiah.fi/interact/>", Tekes–project 40691/00), which is mainly funded by the National Technology Agency, we are designing a general platform for systems which use a natural language interface in communicating with their users. The Interact project is a joint effort between the University of Art and Design Helsinki, the University of Helsinki, the Helsinki University of Technology and the University of Tampere. As a demonstration we are constructing a system which should be able to answer inquiries about the timetables of public transportation.

The problem that we at the University of Helsinki are solving at the moment is the mapping of the utterance of the speaker into relevant semantic units which are in turn processed by the dialogue manager. After dialogue manager has done its processing it produces new semantic units, out of which it is our problem to generate natural language. We are also responsible for creating a dictionary for the speech recognition system.

2 Written vs. spoken Finnish

It is a widely known fact that Finnish is written so that one grapheme corresponds to one phoneme. "Spoken as it is written". Much less known is the fact that spoken Finnish differs greatly from its written

form. The current written form of Finnish was established to serve a compromise between all the Finnish dialects and as such it has never really been a good transliteration of any form of spoken Finnish.

Given the current state of user independent speech recognition, we are forced to use full word lists which include all the morphological forms that we are trying to understand from the user's utterance. Because of the difference between written and spoken Finnish, existing tools cannot handle the word forms used in natural spoken Finnish.

In order to deal with the variation between the several different ways to pronounce any written word, we have to generate a list of the most probable pronunciations for each one. Then we map each one of these pronunciations to standard forms that can be found from the written language. These standard forms are then passed forward to any linguistic analyzers that are being used by the system.

To do this we need a tool which takes as its input the written forms we decide are necessary and as its output gives the forms that can actually be found from spoken language.

3 Two corpora

At the moment we have two small corpora of spoken language dialogues which deal with public transportation. The first corpus includes 32 dialogues and its transliteration was done at the University of Art and Design. The second corpus includes 24 dialogues and it was transliterated at the University of Helsinki. The quality of the first corpus was partly inadequate for the task at hand. The largest problem being the fact that most of the numbers had been transliterated as numerical characters.

The first corpus includes around 8500 words and 2000 different word forms. The morphological analyzer used (*fintwol*, TWOL 92, by K. Koskeniemi and Lingsoft, Inc) understands a little over 80% of the forms and words used.

The statistics for the second corpus are outlined in the table 1.

	# Words	# not understood	% not understood
Client words	2274	610	26.8
HDesk words	2515	514	20.4
All words	4789	1124	23.5
Client forms	846	261	30.9
HDesk forms	773	193	25.0
All forms	1303	378	29.0

Table 1.

37 of all the word forms are such that they do not have any substitute in the lexicon of the *fintwol* (they are mainly proper names). 20 word forms are actually made out of two different lexemes and should not be written together according to the current standards of the written Finnish. After removing these troublemakers the number of different forms is 1246 and out of those 321, or 25,8%, were not understood by the morphological analyzer.

4 Rules of transformation

Generally the variation between the written and the spoken forms seems to be quite regular, and we have been able to construct some rules for generating the spoken forms from the written. For this presentation we are more closely examining the five most important of those rules. When reading the examples it must be noted that several of these rules can be applied to a given word, but so that of the rules one, two and three only one is applied.

Any rule that would change the word into any other existing word with a different

semantic meaning is usually not applied. But this is not always so, for example the word 'että'.

"että" (that) → "ett", by rule 1.

"ett" → "et", by rule 5.

If in the spoken language we see the form 'et', it is most probably a contracted form of the word 'että' and not the written language form 'et' (you do not). This phenomena also creates the problem that the morphological analyzer thinks that it understands the word, even if it doesn't. This percentage decreasing factor has not been considered in the values of the table 1.

4.1 Rule 1

The most common (54/321 instances) variation was the deletion of the final vowel 'a' or 'ä' when it was preceded by a consonant. This variation seems to be fully productive and all words must be thus modified.

/[äa]/ → Ø / C_#

Examples of the rule from the corpus:
 "automaatista" → "automaatist" (from the automat)
 "kyllä" → "kyl" (yes)
 "neljätoista" → "neljätoist" (fourteen)
 "siinä" → "siin" (there)
 "huopalahdessa" → "huopalahdes" (in huopalahti)

4.2 Rule 2

The deletion of the final consonant 'n' (44/321 instances) is fully productive in non-verbal forms.

/n/ → Ø / _#

Examples of the rule from the corpus:
 "ainakin" → "ainaki" (at least)
 "ensimmäinen" → "ensimmäine" (the first)
 "martinlaaksoon" → "martinlaakso" (to martinlaakso)
 "kakkonen" → "kakkone" (number two)
 "tuohon" → "tuoho" (over there)

4.3 Rule 3

The deletion of the final vowel 'i' if it is preceded by the consonant 's' (42/321). This rule is not fully productive and some very short words do not tend to do this, for example the word 'tosi' (true) never becomes 'tos'.

/i/ → Ø / s_#

Examples of the rule from the corpus:

"anteeksi" → "anteeks" (sorry)
 "kuusisataa" → "kuussataa" (six hundred)
 "saisi" → "sais" (would get)
 "uusi" → "uus" (new)
 "kuukausilippu" → "kuukauslippu" (ticket for a month)

This rule could create problems, for it is productive also inside compound words. The lexicon of the speech recognizer is done in such a way that all compound words are divided to non compound words, which solves this problem.

4.4 Rule 4

The deletion of the phoneme 'i' in diphthongs that are further in the word than the first syllable (31/321).

/i/ → Ø / VC⁺V_

Examples of the rule from the corpus:

"aikaisemmin" → "aikasemmin" (before)
 "tarkoitan" → "tarkotan" (I mean)
 "kysyisin" → "kysysin" (I would ask)
 "silloin" → "sillo" (at that time)
 "viimeinen" → "viiminen" (the last)

4.5 Rule 5

If after applying any of the preceding rules (mainly the rule number one) a double consonant is found from the end of the word it shortens to a single consonant (18/321).

/C₁C₁/ → C₁ / _#

Examples of the rule from the corpus:

"missä" → "mis" (where)
 "mutta" → "mut" (but)
 "kehällä" → "kehäl" (at the ring)
 "vaikka" → "vaik" (though)
 "sitten" → "sit" (then)

5 Implementing the rules

The program that does this variation can be fairly small. Table 2 shows the simple substitutions that are needed as regular expressions in Perl.

s/([[^] eyuioääöa])[ää]\$/s1/;
s/n\$/;/
s/si\$/s/;
s/([[^] eyuioääöa])[[^] eyuioääöa]+[eyuoääöa]i/\$1/g;
s/ss\$/s/;
and so on in the format: s/C ₁ C ₁ \$/C ₁ /;

Table 2.

It should generate the original form and all the different forms that any combinations of the above rules can generate. Out of the 321 forms that were originally not understood, these rules would touch 156 (48,6%) and would completely generate 126 (39,3%). Thus reducing the overall percent of the forms not understood to 15,0% from the initial 25,8%. Implementing additional rules will still reduce this percentage.

We have also partly implemented these rules using Xerox's tools for two-level morphology (*twolc*, *Two-Level Compiler 3.1.4 (7.3.8)*). It is easier to handle and to represent the order and parallelism of the rules.

The reason not to include these forms or transformations directly into the lexicon of the morphological or syntactical analyzer is that we don't want to tie our hands into using any one existing software. Also if new analyzers are developed, they will probably be designed for written language.

At the moment we are using Lauri Carlson's *C-parse* as syntactic analyzer. The *C-parse* is designed to analyze *fiⁿtwol*'s output, but for example *TextMorfo* (v.2.0, Kielikone Oy 1999) is not. If in the future we decide to use *TextMorfo*, it is important that the mapper from spoken to written language is an independent module.

The differences between the syntax of the spoken and written languages is a

completely separate problem of equal or greater proportions and is not discussed here.

Table 3 shows some examples from the generated lexicon for the speech recognizer.

MISSÄ <i>m i s s a e / m i s s / m i s</i>
SILLOIN <i>s i l l o i n / s i l l o i / s i l l o / s i l l o n</i>
TARKOITAN <i>t a r k o i t a / t a r k o t a / t a r k o i t a n / t a r k o t a n</i>
UUSI <i>u u s i / u u s</i>
VIIMEINEN <i>v i i m e i n e n / v i i m e i n e / v i i m e n e n</i>

Table 3.

6 Further use for the rules

The use of these rules for other purposes such as speech generation has been contemplated. It would be a significant improvement to the quality of any speech synthesizer if it were able to generate word forms that are found in actual spoken language. Written language word forms

have not been considered a real problem because of the low quality of the other more important attributes like intonation and stress in the current synthesizers. The USIX Suopuhe project is developing a synthesizer with better intonation and stress. The use of the above rules especially with numerical expressions is considered.

References

- Koskenniemi, Kimmo. 1983. *Two-level morphology : a general computational model for word-form recognition and production*. Helsinki, University of Helsinki.
- Lehikoinen, Laila. 1994. *Suomea ennen ja nyt : suomen kielen kehitys ja vaihtelu*. Helsinki, Finn Lectura.
- Suihkonen, Pirkko. 1988. *Murteiden generointia atk:n avulla : showmur-ohjelma*. Helsinki. University of Helsinki.
- Wall, Larry. 2000. *Programming Perl*. Sebastopol (CA), O'Reilly.