Finite-state morphology and information retrieval

Kimmo Koskenniemi
Department of General Linguistics
FIN-00014 University of Helsinki
kimmo.koskenniemi@helsinki.fi

Abstract. A source of potential systematic errors in information retrieval is identified and discussed. These errors occur when base-form reduction is applied with a (necessarily) finite dictionary. Formal methods for avoiding this error source are presented, along with some practical complexities met in its implementation.

1 Introduction

There are certain morphological tasks which are relevant to Information Retrieval but which do not seem important when morphological analysis is considered in general. In this paper, we study the effects of morphological analysis in a framework of full-text retrieval. We, thus, assume that a set of documents is being accessed according to individual words occurring in them. The language in the documents is assumed to be inflecting to the extent that some morphological processing is needed.

If the search were based on the keywords in their base form, and the index would contain the unprocessed word-forms, lots of the occurrences of the keywords would be lost. Plain truncation helps, but does not solve the whole problem in all cases. Especially the use of truncation is virtually useless, if inflectional elements are prefixed, as is the case in Semitic languages (such as Arabic), or in Bantu languages (such as Swahili). Even apparently easier languages, with suffixation alone, the word stems may be subject to alternations, and then the truncation is not an efficient solution.

2 Search stems

Document retrieval in languages which are exclusively suffixing, can be approached by using so called *search stems*. This method compensates for the possible stem alternations, and it consists of producing a set of prefixes which cover all possible inflected forms of the lexeme. This is done without specific reference to a dictionary, i.e. based on the productive rules of the language. E.g. the search stem generator for a Finnish would give the following results out of a noun *"kauppa"* ('a shop'):

kauppa-
kaupa-
kauppo-
kaupo-

It has proven to be possible to write such search stem generators for several languages, and some of them have been in production use for more than ten years [8].

The use of this method has no effect on the number of distinct word-forms in the so called inverted file. On the other hand, it monotonically improves the recall. It fails, however, to retrieve compounds where the key word is in any other position except the first.

With inflecting languages such as Finnish, the index file (which is also called the inverted file) tends to grow along with the size of the text database. With English text data bases, it is said that the number of different word-forms stops growing at a certain point. No similar saturation has been observed with Finnish texts. Thus, this method is bound to be less than optimal in terms of the space requirements.

3 Analysis without a lexicon

The morphological processing could, in principle, be done without a lexicon, i.e. based exclusively on rules of inflection so that instead of the set of actually possible base forms, one produces a larger set of base forms where all potential and theoretical possibilities are given as well. E.g. in Swedish, such an analyzer without a lexicon would be forced to analyze a word-form *"pojkar"* at least in four ways:

"pojke" N UTR INDEF PL NOM
"pojka" V ACT PRES
"pojka" N UTR INDEF SG NOM
"pojkar" N UTR INDEF SG NOM

Only the first of these is a real alternative, others are nonexistent.

We will elaborate this method briefly in this paper in another perspective, and it suffices to note at this stage, that morphological analysis without a lexicon is likely to produce lots of spurious base-forms. This would be undesirable and inefficient in terms of space requirements. Such a method would possibly also be suboptimal in terms of the execution speed when retrieving documents.

4 Analysis with a lexicon

It is more common to perform the base-form reduction using a fixed lexicon. For a given word-form, exactly those base

Proceedings of the ECAI 96 Workshop
Extended Finite State Models of Language
Edited by A. Kornai.

forms will be produced which the lexicon permits by including an appropriate entry (or a mechanism for compounding or deriving the complex stem). Thus, the set of possible base forms are strictly defined and restricted in advance. This is an advantage when we consider the bulk of the word-forms in ordinary documents which are, indeed, covered by conventional, and electronic dictionaries.

It is well known that the lexical inventory of normal unrestricted documents is neither fixed nor bounded. We may and will encounter more and more proper names, new terms, acronyms as we extend our collections of texts. This problem is routinely approached by so called *morphological heuristics*, or *morphological guessing*. These are algorithms which try to deduce the possible base forms of the word-forms which could not be recognized by the main morphological analyzer (e.g. as part of the ENGCG parser of English [10]). In fact, this is roughly what the morphological analysis without a lexicon is, as introduced above. The typical difference between those two would be that the morphological heuristics ignores the exceptional and closed inflectional classes covered by the main lexicon, and concentrates in describing the inflection of names, foreign loan words, newly coined words etc.

5 The problem

One problem remains, even after we have implemented a morphological analyzer and augmented it with a morphological heuristics module. The problem arises from the combinatorics: some word forms which receive analyses from the morphological analyzer might be ambiguous in the following way:

- the word-form receives an analysis and a base form by the lexicon, and
- the true interpretation of the word-form is actually another lexeme, not in the lexicon.

Proper names can have almost arbitrary shapes, and we are currently exposed to news and texts from all parts of the world. Trade marks and various acronyms are being invented constantly. As an example, a Russian politician, Mr. Boris Pankin, was in the news for some time, and his name happens to be homographic to a Finnish word *"pankki"* ('a bank') in genetive singular. If we would be using the word-form reduction into base forms using any reasonable lexicon of Finnish, some inflectional forms (including the most common one, the nominative singular) would be associated with a wrong keyword *"pankki"*. As a result, these instances would not be retrieved with a query concerning *"Pankin"*. The same thing happens through compounding, eg. in Finnish the word-form *"autonomia"* ('authonomy') could be either of the following:

"autonomia" Noun Nominative Singular
"auton_oma" Noun Genetive Singular + Adjective Positive Partitive Plural

The second analysis ('car' Genetive + 'owned by' Partitive) is nonsense but it would be the only analysis if there were no entry for ('authonomy') in the dictionary.

It is easy to see that it is rather hard to do anything for this particular problem in advance, i.e. while indexing the documents. We ought to suspect every single word-form in the input, and it is not easy to see how the problem instances could be identified.

But, during the actual queries, we can easily detect whether a keyword was one of the known ones, simply by applying the same morphological analyzer to the keyword. If it is analyzed as a base form of itself, then it is a known word, and otherwise it is not. This seems to be a bit too late because the indexing has already been completed (and has made the error) because of the lack of this knowledge.

The essential question during the retrieval is: under which base-forms did our lexicon put the possible inflectional forms of the given keyword. In other words, we should

find all base-forms in our lexicon
which have inflectional forms
in common with the given keyword.

There is a known, straight-forward solution to circumvent this problem by using two index files, one containing the unmodified word-forms, and the other consisting of the base forms produced using a fixed lexicon. In this scheme, the known query words are retrieved through the base-form index file, and the unknown keywords through the word-form index file (using a search stems). This approach is not discussed here any further because it is obviously uneconomic in terms of its space requirements.

6 Finite state representation

Morphological analysis with a lexicon has been described in terms of finite-state machines and regular relations for some time. Kaplan and Kay proposed and implemented a scheme of cascaded finite-state transducers in 1980ies [1]. Koskenniemi designed his two-level model based on this work [7, 9, 5], and there is a host of full scale morphological analyzers available now, based on this framework. Later on Karttunen generalized this scheme into a model where more than two levels could be permitted, and where the lexicon is implemented directly as a bidirectional finite-state transducer [3, 4]. Their approach is followed here as the formal framework of the representation.

7 Fixed lexicon transducer

The morphological analyzer using a fixed lexicon can, thus, be expressed in terms of a finite-state transducer, or a regular relation R. This relation is not one-to-one because word-forms may be ambiguous. It is not defined to all strings of the alphabet either, as ungrammatical word-forms have no analysis. We study here a special version R_L of this relation: one which associates pairs of strings consisting of:

- a word-form ω which is a string over the alphabet Σ
- a base-form λ (which is a string over the alphabet Σ) appended with the part of speech code γ (which is a one symbol long string of Γ).

We have thus

$$(\lambda\gamma) R_L (\omega)$$

if and only if ω is a valid inflectional form of the base form λ which has the part of speech γ.

8 Lexicon transducer for morphological heuristics

We will make use of a (theoretically possible) transducer for the open lexicon or the morphological heuristics alias morphological guessing which we call R_H. We try to schetch the definition this transducer, and assume the mathematical existence of it when proposing a solution for the main problem of this paper.

In this version of the lexicon transducer, we replace the fixed list of stems with a general expression which is open enough to cover all anticipated proper names, foreign words, new terms, acronyms etc. which could occur in texts. It is worth while to note that only productive inflectional patterns need to be included in R_H. Exceptions and idiosyncratic patterns are not likely to occur in new words.

For certain types of languages, it is not particularly difficult to express R_H if we already have a formulation of the R_L at hand. If no stem variations are associated with the inflectional class, then an expression such as

$$\Pi_0 = \{(\sigma : \sigma)^* : \sigma \in \Sigma\}$$

could be inserted in the lexicon (and at the same time, all existing entries in that class could be removed).

If stem final alternations occur in a class, then the Π_0 inserted in place of the (truncated) stems to be followed by the existing set of alternating stem ends (a so called minilexicon) handling the alternation, and these stem ends, in turn would be followed by the actual inflectional endings (as in the standard lexicon).

If morphophonemes are anticipated to occur in the stems, they have to be properly embedded in the expressions and surrounded by 'wild cards', and properly chained to sets of inflectional endings.

We tentatively assume that a relation R_H can be expressed following such lines, and using e.g. the LEXC formalism (see [2] for more details).

There is one serious reservation to this simple scheme. The regular expression used as the building block in the above construction is of the type:

$$Y = \Sigma^* X$$

where X is an expression corresponding to a sizable finite-state machine. The deterministic version of automaton corresponding to Y tends to be (exponentially) large. We cannot, thus, claim that the concrete compilation of the run-time R_H is feasible.

9 Theoretical solution

These two transducers, if available, would provide us with an answer to the main question presented above. Let us suppose that a given keyword $\lambda\gamma$ is not defined by the fixed lexicon. If we would produce the (possibly very large) set of word-forms ω through the transducer R_H^{-1} and analyze them with the transducer R_L we would get the set of base forms

$$\lambda_1\gamma_1, \ldots, \lambda_k\gamma_k$$

for which some inflectional forms coincide with some inflectional forms of the $\lambda\gamma$.

The transducers involved here are constant, not depending on the keyword $\lambda\gamma$. Thus we could avoid the (often impossible) task sketched above by computing the transducer corresponding to the composition:

$$R_C = R_L \circ R_H^{-1}$$

This relation

$$(\lambda_i\gamma_i)R_C(\lambda\gamma)$$

holds if and only if λ and $\lambda_i\gamma_i$ have common inflectional forms. In other words, the formula would look like:

$$(\lambda_i\gamma_i)R_C(\lambda\gamma) \Leftrightarrow \exists\omega \in \Sigma^* : ((\lambda_i\gamma_i)R_L(\omega) \wedge (\omega)R_H(\lambda\gamma))$$

It can be speculated that this transducer could be reasonable in terms of its size because it is constrained by the existing fixed lexicon R_L. However, as we noted before, R_H is likely to be ill behaving in this respect.

A conjecture is, then, that the transducer R_C could be computed more feasibly by reversing the components, i.e. by constructing R_H as a reversed transducer $R_H^{\leftarrow}$ which is composed with the reversal of the fixed lexicon transducer:

$$R_C = (R_L^{\leftarrow} \circ (R_H^{\leftarrow})^{-1})^{\leftarrow}$$

10 Future work

The scheme which was presented in this paper has been verified in a rather small scale using the Xerox Finite State Calculus [6]. It remains to be implemented with a full scale description of some example language.

We have not discussed the full complexity of dealing with unknown keywords. More work is needed e.g. to elaborate general and comprehensive methods to construct the transducer R_L for all types of languages. Furthermore, various more complex cases, e.g. where one non-initial component of a compound is missing from the dictionary remain to be elaborated in more detail.

It is likely that there are other paths along which R_C could be built. E.g. one could try refrain from determinizing R_H before the composition.[1] The strategy presented in this paper might work when using the current LEXC and IFSM, but this might require some careful planning and/or preprocessing of the inflectional information in the fixed lexicon.

REFERENCES

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

[2] Lauri Karttunen, 'Lexicon compiler', Technical report, Xerox, Palo Alto Research Center, California, (1993).

[3] Lauri Karttunen and Kenneth Beesley, 'Two-level rule compiler', Technical Report ISTL-92-2, Xerox, Palo Alto Research Center, California, (1992).

[4] Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen, 'Two-level morpholgy with composition', in *COLING-92*, pp. 141–148, (1992).

[5] Lauri Karttunen, Kimmo Koskenniemi, and Ronald Kaplan, 'A compiler for two-level phonological rules', in *Tools for Morphological Analysis*, 1–61, Center for the Study of Language and Information, Stanford University, California, (1987).

[1] As suggested by Pasi Tapanainen, (personal communication).

[6] Lauri Karttunen and Todd Yampol, 'Interactive finite-state calculus', Technical report, Xerox, Palo Alto Research Center, California, (1993).

[7] Kimmo Koskenniemi, *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*, Publications, No. 11, University of Helsinki, Department of General Linguistics, 1983.

[8] Kimmo Koskenniemi, 'Finstems: A module for information retrieval', in *Computational Morphosyntax: Report on Research 1981–84*, ed., Fred Karlsson, 81–92, University of Helsinki, Department of General Linguistics, Publications, No. 13, Helsinki, (1985).

[9] Kimmo Koskenniemi, 'Compilation of automata from morphological two-level rules', in *Papers from the Fifth Scandinavian Conference on Computational Linguistics, Helsinki, December 11–12, 1985*, ed., Fred Karlsson, 143–149, University of Helsinki, Department of General Linguistics, Publications, No. 15, Helsinki, (1986).

[10] Atro Voutilainen, 'Experiments with heuristics', in *Constraint Crammar: A Language-Independent System for Parsing Unrestricted Text*, eds., Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, 293–314, Mouton de Gruyter, Berlin, New York, (1995).