A Knowledge Acquisition and Management System for Morphological Dictionaries

Pius ten Hacken°, Stephan Bopp*, Marc Domenig°, Dieter Holz°, Alain Hsiung°, Sandro Pedrazzini¶

°Universität Basel, IFI
Petersgraben 51
CH-4051 Basel

*Vrije Universiteit Amsterdam
SR Lexicologie
NL-1081 HV Amsterdam

¶IDSIA
Corso Helvezia 36
CH-6900 Lugano

Abstract

A system for the acquisition and management of reusable morphological dictionaries is clearly a useful tool for NLP. As such, most currently popular finite-state morphology systems have a number of drawbacks. In the development of Word Manager, these problems have been taken into account. As a result, its knowledge acquisition component is well-developed, and its knowledge representation enables more flexible use than typical finite-state systems.

1. Introduction

It has long been recognized that there is a need for reusable dictionaries for NLP-systems. A system for the development and maintenance of such dictionaries can be viewed as a knowledge acquisition and management tool. The major objectives from this perspective are support for efficient dictionary construction, and maintenance of consistency within the database.

As a starting point, it makes sense to take morphological dictionaries of a type that can at least solve the *mapping problem* between text words and dictionary entries. This problem directly involves inflection, (graphic) clitics (e.g. German *zum* = *zu dem*), and multi-word units, while wordformation is necessary for the recognition of newly coined words. Concentrating on these morphological dictionaries recommends itself because the information they contain is relatively well-understood (compared to e.g. semantics), can be isolated fairly well from more application-specific types of knowledge, and is necessary for almost every NLP-application.

In this project note, we will first consider finite-state morphology systems from the perspective of knowledge acquisition and management, and observe some shortcomings (section 2). Then Word Manager (WM) is presented, and it is shown how WM avoids these drawbacks (section 3). In section 4, an overview of the present state of the WM-project is given.

2. Finite-State Morphology Systems

One of the most widespread approaches to morphological dictionaries is finite-state morphology. Currently, the most attractive architecture is as in Figure 1 (Karttunen et al. (1992)):

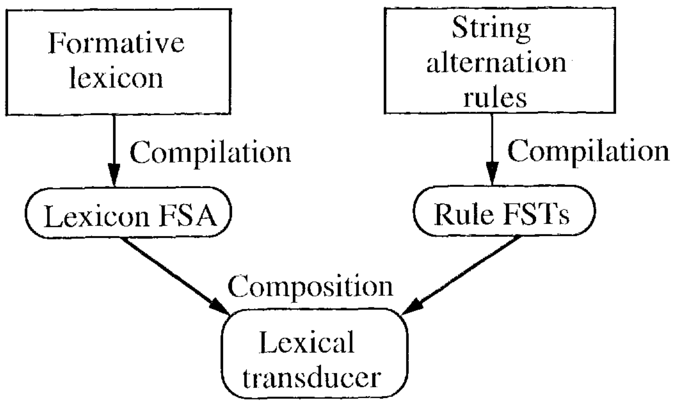

Figure 1: Architecture of Two-Level Morphology.

Compared to the original systems of Koskenniemi (1983) and Karttunen (1983), the major improvements made during the past ten years were in the compilation of the finite-state transition tables, and the switching from analyzers to transducers. By pushing finite-state technology to its limits, the resulting finite-state machines are extremely fast.

If we consider current finite-state morphology systems as a potential solution to the problem sketched in section 1, i.e. the acquisition and maintenance of morphological and lexical knowledge, they have a number of shortcomings. First, knowledge acquisition is not supported very well: the editing of the formative lexicon and the string alternation rules must be done with text editors. Hence, it is a major difficulty for the expert to conceive the lexicon's structure and the string alternation rules. In particular, the system provides little support for the visualization of the structuring and interaction between the entities specified. Secondly, the knowledge is represented specifically for the purpose of mapping from text words to their analysis and vice versa. Flexibility in this respect is lacking. Thirdly, the mapping problem as defined above is only partially covered: neither multi-word units, nor compounding and prefixation can be handled appropriately with the basic systems.

3. Word Manager

The three drawbacks of finite-state morphology mentioned have served as a starting point in the development of Word Manager (WM). The resulting properties of WM are presented in the three following subsections.

3.1. Coverage

A strictly finite-state mechanism has a number of problems in covering natural language morphology, as has been recognized earlier (e.g. Kay (1987)).

In order to treat prefixation and compounding in a way parallel to suffixation, the rules for combining formatives in WM are context-free. Another difference with two-level morphology is the basic distinction between inflection and wordformation in WM. Inflection is treated as the paradigmatic realization of certain features on a lexeme, whereas wordformation is the application of a rule to a lexeme, resulting in a new lexeme. This type of distinction is linguistically motivated and pragmatically elaborated by ten Hacken (1993).

The rules for string alternation in WM are similar in function to two-level rules, but two additional ways are offered to restrict their domain of application. Whereas two-level rules can only see and change strings, WM string alternation rules can also see (but not change) features. Besides they can be defined for individual classes of lexemes or for individual entries, for the treatment of exceptions. The entire formalism of this part of WM is described in Domenig & ten Hacken (1992).

The subsystem Phrase Manager, described in Pedrazzini (1994), covers all cases where the mapping between text words and lexemes is not one-to-one. This includes (graphic) clitics and multiword units. The clitics mechanism may split up text words and rearrange the parts before further analysis. Multiword units are recognized and assigned a structure on the basis of the string, and treated as possible analyses alongside the literal ones.

Together, the mechanisms included in WM cover the entire mapping problem.

3.2. Compilation

The WM formalism can be compiled in a variety of ways. As opposed to compilation of current finite-state systems, the output is not restricted to an analyzer or transducer.

The basic compilation converts rules and entries into a network structure recording links between rules, formatives, and lexeme entries. On the basis of this structure, WM can offer dynamic knowledge aggregation options, to be used for browsing, debugging (rule application tracing) etc.. Another use of this structure is in the access to the data by client applications, where any type of view on the data is supported.

By compiling the wordformation rules into a unification-based grammar, WM can analyze unknown words, and, drawing from the information gathered in the parse tree, generate their lexeme entries (including all inflectional forms of the new entry). Thus, the wordformation rules can be used for semiautomatic construction of the dictionary.

In principle, there is no limit to the types of different formats and rules which can be generated out of the basic network structure. It is straightforward, for example, to generate the input for Koskenniemi's two-level system (Koskenniemi (1983)) or Karttunen's lexical transducer system (Karttunen et al. (1992)).

3.3. Knowledge Acquisition

In the construction of a WM dictionary, linguistic and lexicographic knowledge are separated. As opposed to coding in finite-state morphology, it is not a single person who has to keep track of all the rules, know all special symbols and write all the entries. Instead, tailor-made interfaces for linguistic and lexicographic experts have been developed.

The linguist's interface supports the description of the morphological rule system of a language in the WM-formalism. Besides formulating the rules themselves, the linguist has to give at least one example per rule, and to describe all exceptions.

The compilation step checks syntactic correctness and signals a number of semantic errors or probable errors. The compiled database can be inspected to see if the rules yield the desired results. The interface offers an incremental compilation facility, so that it is possible to get an inspectable version of the database without entirely recompiling it. This makes a gradual expansion of the rule system with frequent testing of intermediate results feasible. Thus, the way compilation has been implemented in WM contributes to a higher degree of consistency.

In the lexicographer's interface, the rule-base coded by the linguist is presupposed and two tasks are supported:

- New formatives (stems) can be specified and linked to existing regular inflection rules.
- Existing formatives can be combined in novel ways, according to existing rules for wordformation and multiword units.

Obviously, the interface can show the consequences of any lexicographic decision for the forms generated by the rules concerned. More active support is also provided, so that the system proposes an analysis interactively. For the first task, an inflection rule is proposed on the basis of forms given by the lexicographer. In the second task, the support consists of parsing the input on the basis of the rules in the database. The lexicographer can choose and inspect one of the proposals, confirm it as it is, or go back and choose a different proposal. The resulting entry is incrementally added to the lexical database.

The lexicographer's interface helps the lexicographer concentrate on lexicographic decisions, rather than problems of encoding them. Since wordformation accounts for a major portion of a language's vocabulary, and this part of a WM-database is constructed by rule application a high degree of consistency is guaranteed.

4. Present State of the Project

WM has been fully implemented as a client/server system, where the server runs on either Sun workstations or Macintosh computers; the client currently runs on Macintosh only. The implementation includes Phrase Manager, the compiler, and the linguist's and lexicographer's interfaces as described above. A comprehensively documented version has been published on an ftp server (ftp@ifi.unizh.ch).

A full rule database, including morphological and phrasal rules, has been implemented for German, and complete morphological rule databases for Italian (described by Bopp (1993)) and English. The development of these databases has shown that the expressive power of the WM-formalism is sufficiently rich for at least some of the languages most frequently used in NLP-systems currently under development.

5. Conclusion

From the perspective of knowledge acquisition and maintenance, WM has a number of properties that distinguish it from current finite-state morphology systems in a positive way. It has a larger coverage, including compounding, prefixation, and multi-word units; compilation is much more generic in that it enables flexible use of information in the database; and knowledge acquisition is supported by tailor-made interfaces for linguists and lexicographers. Therefore, WM is a tool that may be seen as an ideal supplement for today's finite-state analyzers and transducers and thus a further step towards the development of truly reusable dictionaries.

References

Bopp, Stephan (1993), *Computerimplementation der italienischen Flexions- und Wortbildungsmorphologie*, Olms, Hildesheim.

Domenig, Marc & ten Hacken, Pius (1992), *Word Manager: A System for Morphological Dictionaries*, Olms, Hildesheim.

ten Hacken, Pius (1993), *Defining Morphology: A Principled Approach to Determining the Boundaries of Compounding, Derivation, and Inflection*, Ph.D. Dissertation, University of Basel, to appear at Olms Verlag, Hildesheim.

Karttunen, Lauri (ed.) (1983), 'KIMMO: A Two Level Morphological Analyzer', in *Texas Linguistic Forum* 22, Department of Linguistics, University of Texas, Austin, p. 163-278.

Karttunen, Lauri, Kaplan, Ronald M. & Zaenen, Annie (1992), 'Two-Level Morphology with Composition', *Proceedings of Coling-92*, p. 141-148.

Kay, Martin (1987), 'Nonconcatenative Finite-State Morphology', in *Proceedings of the Third Conference of the European Cahpter of the Assocation for Computational Linguistics*, Copenhagen, p. 2-10.

Koskenniemi, Kimmo (1983), *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*, University of Helsinki.

Pedrazzini, Sandro (1994), *Phrase Manager: A System for Phrasal and Idiomatic Dictionaries*, Olms, Hildesheim.