THE SELF-EXTENDING LEXICON: OFF-LINE AND ON-LINE DEFAULTING OF LEXICAL INFORMATION IN THE METAL MACHINE TRANSLATION SYSTEM (1)

Geert Adriaens

Siemens CSL &
University of Leuven
Department of Linguistics
METAL Project
M. Theresiastraat 21
B-3000 Leuven, Belgium
siegeert@cs.kuleuven.ac.be
+ 32 16 285091

Maarten Lemmens

University of Leuven
Department of Linguistics
Blijde-Inkomststraat 21
B-3000 Leuven, Belgium
+32 16 284776

0. INTRODUCTION

The DEFAULTER component of the METAL MT system has been developed as a tool for extending the existing lexicons (off-line defaulting) and is the basis for a category-guessing device for unknown words (on-line defaulting).

1. SOME BACKGROUND FOR THE METAL MT SYSTEM (2)

1.1. METAL MONOLINGUAL DICTIONARIES

METAL monolingual lexicon entries are represented as feature-value structures, accessible by their citation or "canonical" form. For each canonical form, the allomorphic variants (spelling variants, irregular forms, etc.) are stored.

1.2. MORPHOLOGY AND MORPHOSYNTACTIC (ANALYSIS) RULES

In METAL, morphological analysis is a recursive process of lookup and segmentation that scans input words from left to right in search of their component parts. This results in a set of possible interpretations which correspond to acceptable sequences of morphemes recognized in the word (3). Words (or parts of complex words) which are not in the dictionary will be assigned the category UNK (for UNKnown). The morphemes that are the result of morphological analysis are then put in a chart structure for further processing by (morpho)syntactic rules.

2. OFF-LINE DEFAULTING

2.1. GENERAL DESCRIPTION

The defaulter first checks whether a word is in the dictionary (level 0). If not, it tries to find morphologically related entries, so that the information for the new words can be taken from those existing entries (level 1). If no related entries can be found, the form of the word can give indications of its (mainly) phonological and morphological characteristics (level 2). Hence, the need to organize this knowledge in an exhaustive, modular and easily extendable way, so that at least part of the information for new entries can be generated automatically.

2.2 DETAILED DESCRIPTION

The DEFAULTER system consists of three modules:

(1) a BASIC module containing language-independent functions (like table manipulation, dictionary checking, creating defaulted entries in METAL format, general string manipulation, etc.). Furthermore, the basic module contains the necessary information about what features (of the set defined for METAL) should not be copied from entries that are already in the dictionary, but should get new values for the particular word in question.

(2) for each language, a language-dependent module containing functions whose algorithms depend on the language involved.

(3) for each language, a set of tables containing language-dependent information in a declarative way. The smartness of the system depends largely on their completeness and degree of refinedness. There are three major types of tables:

(3.1) STANDARD-ENTRIES-TABLES, containing for each category the minimal feature-value information that has to be in the lexicon.

(3.2) CONTROL-TABLES, containing for each category the functions to be applied for trying to find a related root form in the lexicon.

(3.3) ENDINGS-TABLES, containing for each category defaulted the endings that allow one to fill in the values for specific features (see Lemmens 1988). An entry in the table has the following general structure:

```
(ENDING-PATTERN
 (ALO-PATTERN1 (FEAT1 (VAL1 .. VALm)
                ..
               FEATn (VAL1 .. VALn)))
 (ALO-PATTERN2 (FEAT1 (VAL1 .. VALm)
                ..
               FEATn (VAL1 .. VALn)))
 ...
 (ALO-PATTERNn (FEAT1 (VAL1 .. VALm)
                ..
               FEATn (VAL1..VALn))))
```

(3.4) beside these three major tables, the system needs to know about the linguistically motivated ways to find the root form of a morphologically complex word. For verbs, nouns, adjectives and adverbs (subject to productive morphological processes), the system has exhaustive lists of derivational prefixes it will try to match with the word to be defaulted. If these prefixes require that certain defaulted values be changed, this will be stored in additional conversion tables for overriding default information (4).

Off-line defaulting plays a major role in the INTERCODER subsystem, a window and menu-based interactive coding tool that hides the internal representation of information in the lexicons from the user and presents it in a more friendly way. Secondly, developers of the METAL system can simply default files with words and create a new file with defaulted entries. These files can then be edited with any type of editor to correct and complete the entries before adding them to the lexicons.

2.3. PROBLEMS WITH OFF-LINE DEFAULTING

Most problems with off-line defaulting occur at level 1, when the word takes over certain features from its morphologically related basic form, while this is incorrect. Unfortunately, these errors are hard to predict. At level 2 (when defaulting can only resort to the endings-tables), errors are mostly a mere consequence of incompleteness in these tables. These errors are usually easier to detect because they are more striking (e.g. when they lead to the creation of several impossible allomorphs for a word).

3. ON-LINE DEFAULTING

3.1. GENERAL BACKGROUND

Instead of resorting to assigning either one single default category (say, noun) to the UNK (the single-category approach), or all open-class lexical categories (the all-categories approach), we tried to develop an intermediate solution, the some-categories approach. The challenge is to find out if the form of a unknown word, inflected or not, can convey crucial categorial information. Even if the attempt at on-line defaulting (using endings information and suffix-stripping) is incapable of disambiguating categorially, at least partial disambiguation may be possible, leaving the system with a minimum of acceptable guesses of a category plus the associated feature-value information for the word involved (noun and verb, for instance).

3.2. ON-LINE DEFAULTING IN METAL: PAST AND PRESENT

3.2.1. SINGLE-CATEGORY DEFAULTING

The earlier on-line defaulting approach consisted of calling a category-guessing function in the test part of three UNK-rewriting morphosyntactic rules, viz. NO -> UNK, ADJ -> UNK, and VB -> UNK. The category-guessing function took the form of the unknown word as input, and returned either NO, ADJ, VB, or NIL, depending on whether it could predict the unknown to be a noun, adjective or verb respectively (using lists of derivational and inflectional suffixes in the process). If the guess-cat function returned NIL, the word was assumed to be a noun (the catchall default). The function applied a simplified right-to-left morphological analysis algorithm, trying to find an acceptable pair of a derivational and an inflectional suffix for a particular category. This approach has a few shortcomings: (1) It is a single-category defaulting scheme: the guess-cat function only returns one guess, and leaves it at that. Furthermore, the guessing process will not be useful for languages with a high degree of categorial ambiguity. (2) Guess-cat only returns the categorial information and no specific feature-value information, whereas the form of the unknown word may reveal much more specific feature-value information. (3) The parser will always try the three UNK-rewriting rules (and call the guess-cat function at least three times with the same string), though only one of the three rules can succeed. Moreover, a possibly morphologically complex word is rewritten into a higher-level node without the grammar knowing about its component morphemes.

3.2.2. SOME-CATEGORIES DEFAULTING

Unfortunately, the ENDINGS-TABLES used in off-line defaulting could not be used in their original form for on-line defaulting. First of all, they are too unspecific to predict the category of the word, and secondly, they rely on the input word being a canonical (citation) form and contain no information about inflectional morphology. Hence, a unique new table had to be constructed that contains not only endings of stem forms, but also inflectional suffixes that allow one to disambiguate an unknown word. Moreover, multiple guesses (two at most) are allowed. The table returns one or more categories plus other feature information.

```
(defvar *DEF-DUTCH-ON-LINE-ENDINGS*
 (def-sort-endings-table
  '(...
    ("iteit" (NST (CAN "*") (ALO "*") (GD F)
                  (CL S-0 P-EN) (DH DE))))
    ("lijks" (AST ((CAN "*") (ALO "*")
                   (CL P-0 P-E))))
    ("ieel" ((NST ((CAN "*") (ALO "*") (GD N)
                   (CL S-0) (DH HET)))
             (AST ((CAN "*") (ALO "*")
                   (CL S-0) (DG SU)))))
     ...
    ("dt" (VST ((CAN "-en")(ALO "t")(CL PR-T))
               (V-FLEX
                  (0 ((CAN t) (ALO T) (CL PR-T)
                         ... )))))
     ...)))
```

The algorithm tries to match the unknown with the endings in the table, gradually stripping off potential inflectional suffixes (as retrieved from the lexicon). The disambiguating potential of these suffixes is also used in this process. If, for example, a word ends in an adjective morpheme and in the endings-table both noun and adjective are listed as possible categories for the string without the morpheme, only the AST category will be defaulted. If the whole strip-and-match process is unsuccessful, the catch-all default remains the noun, which gets all possible values for its features ((NU SG PL) (GD M F N) ...). Instead of invoking category guessing in the grammar rules, we decided to activate the guessing process right after the left-to-right full-fledged morphological analysis has returned an UNK analysis. The guessing process will yield the right lexical categories and put these into the chart. This means that (1) the UNK category disappears as a "lexical" category and (2) all component morphemes of a morphologically complex unknown word are added to the chart with all their associated information. The linguist-developer controls the guessing process through the modularly accessible on-line defaulting table.

3.3. PROBLEMS WITH ON-LINE DEFAULTING

The very nature of the defaulting itself implies that it is not error-free. Still, in many cases the number of exceptions to certain ending strings was rather limited, and mostly they could be accounted for by including a more specific (that is, a longer) ending string in the table. In some cases, such a solution was not feasible, and the exceptions had to be entered into the dictionary.

4. FURTHER RESEARCH

As far as further research into off-line defaulting is concerned, we will be looking at the potential of the approach for defaulting transfer lexicon entries (and not only monolingual ones). For instance, we could suggest a translation for affixed words, if their heads are already in the transfer dictionary. An example can make clear what this means. Suppose the transfer dictionary for translation from Dutch to French contains an entry *gelukkig* -> *heureux (happy)*. Suppose now that we want to default the word *ONgelukkig* (*UNhappy*) in the Dutch monolingual dictionary. If we knew about a correspondence between Dutch *on-* and a French adjective-deriving prefix with the same meaning (say, *mal-*), we could first default monolingual Dutch *ongelukkig* on the basis of *gelukkig*, then look at the transfer for *gelukkig* (*heureux*), and default the monolingual French *malheureux*, as well as the transfer entry *ongelukkig* -> *malheureux*. Of course, such an approach relies heavily on unique mappings of phenomena across languages, which will rarely be the case. For *on-*, for instance, *onjuist* (*incorrect*) does not correspond to **malcorrect*, but *incorrect*. Even in these cases, a translation could be suggested, possibly accompanied by alternative prefixes of the target language with the same meaning.

As to on-line defaulting, the current approach is more or less stable for Dutch and French, but we are still refining the strip-and-match algorithm for optimal results. For the other languages in the set of METAL language-pairs (German, English, Spanish), we will look into the usefulness and the feasibility of some-categories on-line defaulting, and see if interesting tables can be constructed for these languages as well.

NOTES

(1) We are greatly indebted to Michael Thum for his careful documentation of the DEFAULTER system.

(2) See e.g. White 1987, Bennett & Slocum 1988, Thurmair 1989 or Adriaens & Caeyers 1990 for full discussions of the different aspects of the METAL system.

(3) For a full account of the morphological process in METAL, see Loomis 1988.

(4) A typical example of default-overriding for Dutch verbs is the following. If e.g. *gaan* (*to go*) has as one of its morphological characteristics that its past participle is formed with *ge-* (*gegaan;* the feature CL will have as one of its values PP-GE), this information must be overridden for the related verb *vergaan* (past participle is *vergaan* - not *gevergaan*, which means CL must be PP-0). These regularities are stored in the *DEF-DUTCH-VST-CL-CONV* table (defining the necessary morphological class conversions for past participles).

REFERENCES

Adriaens, G. & H. Caeyers (1990) - *Het automatisch vertaalsysteem METAL: van onderzoek tot commercieel produkt.* To appear in *Informatie*, October 1990.

Bennett, W.S. & J. Slocum (1988) - *The LRC Machine Translation System.* In Slocum 1988, 111-134.

Boguraev, B. & T. Briscoe (eds) (1989) - *Computational Lexicography for Natural Language Processing.* Longman, London.

Lemmens, M. (1988) - *A Critical Study of the Defaulters in the Belgian Metal-system, and a Design of a Morphologically Guided Category Guesser.* Master's Thesis in Germanic Philology, Leuven 1988 (written in Dutch).

Loomis, T. (1988) - *Morphological analysis in METAL.* Internal documentation written at Siemens AG, K Systeme AP 323, Munich.

Nebendahl, D. (ed) (1989) - *Expertensysteme.* Teil 2: Erfahrungen aus der Praxis. (Engineering und Kommunikation.) Siemens AG, München.

Nirenburg, S. (ed) (1987) - *Machine Translation. Theoretical and Methodological Issues.* (Studies in Natural Language Processing.) Cambridge University Press, Cambridge UK.

Nirenburg, S. (1989) - *Lexicons for Computer Programs and Lexicons for People.* CMU-CMT Paper.

Slocum, J. (ed) (1988) - *Machine Translation Systems.* Studies in Natural Language Processing. (Revised reissue of a special issue of Computational Linguistics vol. II (1985), nos. 1-3.) Cambridge University Press, Cambridge UK.

Thum, M. (1986) - *Documentation of the METAL DEFAULTER system.* Internal documentation written at the Computer Gesellschaft Konstanz GmbH.

Thurmair, G. (1989) - *Aufgabentyp Linguistik: Projekt METAL.* In Nebendahl 1989, 169-195.

White, J. S. (1987) - *The Research Environment in the METAL Project.* In Nirenburg 1987, 225-246.

Zernik, U. (ed) (1989) - *Proceedings of the First International Lexical Acquisition Workshop* (Detroit, Michigan, August 21).