A tagger/lemmatiser for Dutch medical language

Peter Spyns
University of Gent, Division of Medical Informatics
De Pintelaan 185 (5K3), B-9000 Gent, Belgium
Peter.Spyns@rug.ac.be

Abstract

In this paper, we want to describe a tagger/lemmatiser for Dutch medical vocabulary, which consists of a full-form dictionary and a morphological recogniser for unknown vocabulary coupled to an expert system-like disambiguation module. Attention is also paid to the main datastructures: a lexical database and feature bundles implemented as directed acyclic graphs. Some evaluation results are presented as well. The tagger/lemmatiser currently functions as a lexical front-end for a syntactic parser. For pure tagging/lemmatising purposes, a reduced tagset (not suited for sentence analysis) can be used as well.

1 Introduction

Medical patient reports consist mainly of free text, combined with results of various laboratories. While numerical data can easily be stored and processed for archiving and research purposes, free text is rather difficult to be processed by a computer, although it contains the most relevant information. However, only a few NLP-driven systems have actually been implemented (Friedman and Johnson, 1992).

For Dutch, a prototype covering a larger part of the Dutch grammar and medical vocabulary is under development. This paper focuses on a spin-off — c.q. a contextual tagger/lemmatiser (T/L) — of the lexical component of the Dutch Medical Language Processor (DMLP) (Spyns and De Moor, 1996). A T/L is quite valuable for several kinds of corpus studies concerning the medical vocabulary (co-occurrence patterns, statistical data, ...). For efficient sentence analysis in particular, it is necessary to disambiguate the results of morphological analysis before they can be passed on the parser.

In the following sections, we will describe in detail the different knowledge bases (cf. section 2) and the implementation of the major data structures (cf. section 3). Each section is illustrated by an example or some implementation details. The subsequent section (4) is devoted to the evaluation. The paper ends with a discussion (section 5).

2 Linguistic Knowledge

In essence, the T/L is a generate-and-test engine. All possible morphological analyses of a word are provided (by the database or the word recogniser cf. section 2.1), (**generator**), and the contextual disambiguator (cf. section 2.2), (**test engine**), must reduce as much as possible the potentially valid analyses to the one(s) effectively applicable in the context of the given input sentence [1] .

2.1 Lexical Front-end

The dictionary is conceived as a **full form dictionary** in order to speed up the tagging process. Experiments (Dehaspe, 1993b) have shown that full form retrieval is in most of the cases significantly faster than canonical form computation and retrieval. (cf. also (Ritchie et al., 1992, p.201)). The lexical database for Dutch was built using several resources: an existing electronic valency dictionary [2] and a list of words extracted from a medical corpus (cardiology patient discharge summaries). The already existing electronic dictionary and

[1] Before the actual linguistic analysis takes place, some preprocessing (marking of sentence boundaries, etc.) is done.

[2] This resulted from the K.U. Leuven PROTON-project (Dehaspe and Van Langendonck, 1991)

the newly coded entries were converted and merged into a common representation in a relational database (Dehaspe, 1993a). A Relational DataBase Management System (RDBMS) can handle very large amounts of data while guaranteeing flexibility and speed of execution. Currently, there are some 100.000 full forms in the lexical database (which is some 8000 non inflected forms). For the moment, the database contains for the major part simple wordforms. Complex wordforms nor idiomatic expressions are yet handled in a conclusive manner.

However, since an exhaustive dictionary is an unrealistic assumption, an **intelligent word recogniser** tries to cope with all the unknown word forms (Spyns, 1994). The morphological recogniser tries to identify the unknown form by computing its potential linguistic characteristics (including its canonical form). For this purpose, a set of heuristics that combine morphological (inflection, derivation and compounding) as well as non morphological (lists of endstrings coupled to their syntactic category) knowledge. When these knowledge sources do not permit to identify the unknown forms, they are marked as guesses and receive the noun category.

Actually, a difference is made between the regular full form database dictionary and a much smaller **canonical form dictionary**. The latter consist of automatically generated entries. Those entries are asserted as temporary canonical form lexicon entries and do not need to be calculated again by the recogniser part of the T/L when encountered a second time in the submitted text. A substantial speedup can be gained that way.

2.2 The Disambiguator

The contextual [3] disambiguator of the DMLP is implemented as an "expert-like system" (Spyns, 1995), which does not only take the immediate left and/or right neighbour of a word in the sentence into account, but also the entire left or right part of the sentence, depending on the rule. E.g. if a simple form of the verb 'hebben' [have] appears, the auxiliary reading is kept only if a past participle is present in the context [4] .

[3] We only consider the *syntactic* context.

[4] Unlike in English, the past participle in Dutch does not need to occupy a position adjacent to the auxiliary.

The **rule base** can be subdivided into 21 independent rule sets. A specific mechanism selects the appropriate ruleset to be triggered. Some rulesets are internally ordered. In that case, if the most specific rule is fired, the triggering of the more general rules is prevented. In other cases, all the rules of a ruleset are triggered sequentially. Some rules are mutually exclusive. The rules are implemented as Prolog clauses, which guarantees a declarative style of the rules (at least to a large extent).

The **control mechanism** works with an agenda that contains the position of the words in the input sentence. The position in the sentence uniquely identifies a word (and thus its corresponding (group of different) morphological reading(s)). Every position in the agenda is sequentially checked whether it can be disambiguated or not. If an ambiguous word is encountered, its position is kept on the agenda. For every element of the agenda, all possible binary combinations of the syntactic categories are tried (failure driven loop). To avoid infinite loops (repeatedly firing the same rule that is not able to alter the current set of morphological readings), the same ruleset can only be fired once for the word on the same position during the same pass. As long as the disambiguator can reduce the number of readings and the agenda is not empty, a new pass is performed.

3 Software Engineering

In order to preserve the reusability of the dictionary, an extra software layer hides the **database**. This layer transforms the information from the database into a feature bundle containing the application specific features. The software layer restricts and adapts the "view" (just like the SQL-views) the programs have on the information of a lexical entry . This methods allows that all sorts of information can be coupled to a lexical entry in the database while only the information relevant for a specific NLP-application passes "the software filter". Besides the qualitative aspect, the filter can also affect the quantitative aspect by collapsing or expanding certain entries (e.g. the 1st and 2nd person singular of many verbs constitute the same entry in the database but are differentiated afterwards) or excluding specific combinations after examination of the input.

The **feature bundles** constitute the main datastructure of the T/L itself. They are conceived as Directed Acyclic Graphs, which are implemented as open ended Prolog lists (Gazdar and Mellish, 1989). This "low level" implementation is only known by the predicates that make up the interface. Graph-unification provides a neat and easy way to impose various restrictions. A linguistic restriction can be expressed in terms of feature value pairs, which in turn can be represented as a DAG. This DAG acts as filter towards other DAGs. The DAGs that are unifyable with the "filter DAG" meet the imposed restriction. The only thing to do is to define the appropriate filters. The contextual rules mainly consist of such filter DAGs.

The T/L, able to analyse words lacking from the dictionary, is intended to function primarily as a lexical front-end for the DMLP syntactic analyser (Spyns and Adriaens, 1992). However, as the result of the tagging and lemmatising process consists of feature bundles implemented as DAGs, the output format can be adapted very easily if required (by defining various "format filters"). The output format can be transduced to the format required by the "SAC-tools" of the System Management Tools of the Menelas-project (Ogonowski, 1993). Another filter transforms the output to the format of the Multi-Tale semantic tagger (Ceusters, 1994).

4 Evaluation

In order to assess the performance of the T/L, several data sets were used. A learning set of 1314 tokens (5 reports) from the cardiology department (cardio) should eliminate as much as possible errors due to unknown vocabulary. A new large test set of 3167 tokens of 35 neurosurgical reports was fed to the T/L to see how robust it is when confronted with the vocabulary of a completely new domain. The problem with an application of this type is the trade-off between overkill (a good analysis is injustly discarded) and undershoot (an invalid analysis is kept). The extensive tagset (tagset1) provides all the morphosyntactic information as required by the DMLP parser for sentence analysis, while the reduced tagset (tagset2) consists of 15 categories and 25 specifiers (which gives 43 meaningful combinations). This simplification of the syntactic information greatly improves the results.

Table 1: results of contextual tagging with an extensive tagset (tagset1) versus a reduced one (tagset2) on the cardio and neuro sets

	tagset1	cardio	tagset2	cardio
	1314	100 %	1314	100 %
bad	102		39	
2	129	92.23 %	75	97.03 %
1	1083	82.42 %	1200	91.32 %
	tagset1	neuro	tagset2	neuro
	3167	100 %	3167	100 %
bad	446		276	
2	389	85.91 %	261	91.28 %
1	2332	73.63 %	2630	83.04 %

All the results were manually examined and synthesised (cf. table 1). As soon as even one feature of the complete feature bundle with linguistic information is wrong, the analysis as a whole is considered to be incorrect. All the words that have wrong, lacking, doubtful or more than 2 competing analyses are considered as *bad*. Sometimes, two competing readings could not be disambiguated without semantico-pragmatic knowledge. In addition, we deliberately left some ambiguities pending for the syntactic parser to avoid the danger of overkill (cf. also (Jacobs and Rau, 1993, pp.166–167) on this matter). These cases of "double analysis" are grouped in the "class *2*". The question whether these cases should be considered as bad or correct is left open [5] .

The difference between the results is mainly due to the amount of unknown vocabulary (around 9 % for the cardio set vs. around 18% for the neuro set which results in a difference of 82.42 % vs. 73.63 % and 91.32 % vs. 83.04 %) and the nature of the tagsets (82.42 % vs. 91.32 % and 73.63 % vs. 83.04 %).

5 Discussion

As far as we know, only one T/L for medical English exists (Paulussen and Martin, 1992), which has recently been adapted to medical Dutch and extended with semantic labelling (Maks and Martin, 1996). Most of the T/Ls [6] attain a

[5] Probably, the answer will be different depending on the task of the T/L: "pure" tagging or auxiliary function for the parser.

[6] Cf. (Paulussen, 1992) for a detailed overview and discussion of some T/Ls - including CGC, Tag-

95% - 97% score, although for ENGCG a 99.7 % succes rate is claimed (Tapanainen and Järvinen, 1994). All these taggers use a rather restricted tagset. Therefore, we consider it fair to compare only our results on tagset2 with the scores of the mentioned T/Ls. It must be mentioned as well that word order in medical Dutch can be rather free. Moreover, medical sublanguage sometimes deviates considerably from the standard grammar rules. E.g. determiners can be easily skipped, which enhances the difficulty to distinguish a noun from certain conjugated verbal forms. As a conclusion, we believe that, our T/L performs relatively well and still has potentialities for improvement.

Acknowledgements
Parts of this work were supported by the MENELAS (AIM #2023) (Zweigenbaum, 1995) and DOME (MLAP #63-221) projects (Séroussi, 1995) of the E.U. We also would like to thank Luc Dehaspe for his work on the lexical database (Dehaspe, 1993a).

References

Ceusters W., 1994, *The Generation of MULTI-lingual Specialised Lexicons by using Augmented Lemmatizer-Taggers*, Multi-TALE Delivrable #1,

Dehaspe L. & Van Langendonck W., 1991, Automated Valency Dictionary of Dutch Verbs, K.U. Leuven.

Dehaspe L., 1993a, *Report on the building of the MENELAS lexical database*, Technical Report 93-002, Division of Medical Informatics, K.U. Leuven.

Dehaspe L., 1993b, *Full form retrieval versus canonical form computation of morphological data: a performance analysis*, Technical Report 93-004, Division of Medical Informatics, K.U. Leuven.

Friedman C. & Johnson S., 1992, Medical Text Processing: Past achievements, future directions, in Ball M. & Collen M., *Aspects of the Computer-based Patient Record*: 212 - 228, Berlin: Springer - Verlag.

Gazdar G. & Mellish C., 1989, *Natural Language Processing in Prolog: an introduction to computational linguistics*, Addison-Wesley.

git, Parts, Claws, Dilemma, the Parc Tagger and the "Brill tagger" - as well as (Voutilainen, 1995).

Jacobs P. & Rau L., 1993, Innovations in text interpretation, in *Artificial Intelligence* 63: 143 - 191 .

Maks I. & Martin W., 1996, MULTI-TALE: Linking Medical Concepts by means of Frames, *Proc. of COLING 96*, Copenhagen.

Ogonowski A., 1993, *SAC Manuel Utilisateur*, GSI-ERLI, Internal Report.

Paulussen H., 1992, *Automatic Grammatical Tagging: description, comparison and proposal for augmentation*, U.I.A., Wilrijk (M.A. thesis).

Paulussen H. & Martin W., 1992, DILEMMA-2: A Lemmatizer-Tagger for Medical Abstracts, in *Proc. of ANLP 92*, 141 - 146, Trento.

Ritchie G., Russell G., Black A. & Pulman S., 1992, *Computational Morphology: Practical Mechanisms for the English Lexicon*, MIT Press.

Séroussi B., & DOME Consortium, 1995, *Document Management in Healthcare: Final Report*, DOME Deliverable #D.02, Paris.

Spyns P. & Adriaens G., 1992, Applying and Improving the Restriction Grammar Approach for Dutch Patient Discharge Summaries, *Proc. of COLING 92*, 1254 - 1268, Nantes.

Spyns P., 1994, A robust category guesser for Dutch Medical language, in *Proc. of ANLP 94*, 150–155, Stuttgart.

Spyns P., 1995, A contextual Disambiguator for Dutch medical language, in *Proc. of the 7th Benelux Workshop on Logic Programming*, Gent.

Spyns P. & De Moor G., 1996, A Dutch Medical Language Processor, in *International Journal of Bio-Medical Engineering*, (in press).

Tapanainen P. & Järvinen T., 1994, Syntactic Analysis of natural language using linguistic rules and corpus-based patterns, in *Proc. of COLING 94*, 629 - 634, Kyoto.

Voutilainen A., 1995, A syntax-based part-of-speech analyser , in *Proc. of EACL 95*, Dublin .

Zweigenbaum P. & MENELAS Consortium, 1995, Menelas: Coding and Information Retrieval from Natural Language Patient Discharge Summaries, in Laires M., Ladeira M. & Christensen J., *Health in the New Communications Age*, IOS Press, Amsterdam, 82 - 89 .