Proceedings Informatiewetenschap 1996, 33-40, 1996

Part-of-Speech Tagging of Dutch with MBT, a Memory-Based Tagger Generator

Walter Daelemans, Jakub Zavrel
Computational Linguistics and AI
Tilburg University
P.O. Box 90153, NL-5000 LE Tilburg
{walter.daelemans,zavrel}@kub.nl

Peter Berck
Center for Dutch Language and Speech
University of Antwerp
Universiteitsplein 1, B-2610 Wilrijk
peter.berck@uia.ua.ac.be

October 28, 1996

Abstract

We present a part of speech tagger (morphosyntactic disambiguator) for Dutch, constructed by means of the Memory-Based Tagger generation method. In this approach, inductive learning methods are used to derive a tagger, lexicon and unknown word category guesser fully automatically from a tagged example corpus. Advantages of the approach are (i) fast tagger development time without linguistic engineering, (ii) accuracy better than or comparable to state of the art statistical and rule-based approaches, (iii) fast tagging speed, and (iv) reliable unknown word category guessing without the overhead of morphological analysis.

1 Introduction

A Part-of-Speech tagger annotates the words in a text with their morphosyntactic categories. A good tagger is instrumental in a large number of information technology solutions. It can produce a shallow, but accurate linguistic analysis of texts, and therefore features as a central component in many text processing and language engineering applications (ranging from text-to-speech over parsing to information retrieval and document analysis). In order to process large volumes of text, a tagger needs to be robust, fast, and applicable to unrestricted vocabulary text.

The problem of part of speech tagging (morphosyntactic disambiguation) is the following: given a text, provide for each word in the text its contextually disambiguated part of speech (morphosyntactic category). I.e. transform a string of words into a string of tags. E.g., the sentence "`The old man the boats .`" should be mapped to "`Det Noun Verb Det Noun Punc`". The target category inventory (tag set) may range from extremely simple (order 10) to extremely complex (order 1000). Tagging is a hard task because of the massive ambiguity in natural language text. E.g. in the example above, `man` can be both a noun and a verb, context determines that in this case it is a verb. The correct category of a word depends on both its lexical probability $Pr(cat|word)$, and its contextual probability $Pr(cat|context)$. Integrating these two sources of information is the main problem for part of speech taggers.

The tagset and training corpus used differ from one application to the next, and making a tagger by hand is expensive and difficult. Therefore, fast, robust and accurate taggers which can be automatically learned from small annotated example corpora are a commercially interesting information technology product.

Different approaches to tagging have been reported in the literature: stochastic (e.g. Church, 1988; Cutting et al. 1992) and rule-based (e.g. Brill, 1992; Karlsson et al., 1995) methods dominate the field. In this paper, we present a memory-based learning approach to tagging which combines the attractive properties of stochastic and rule-based taggers, and apply it to tagging for Dutch. Our system is a tagger generator; it can be applied to any annotated training corpus, and it yields a working tagger that can accurately annotate previously unseen text in the same fashion as in the training corpus. For this purpose, a lexicon and a disambiguator for known and unknown words are derived fully automatically from the tagged example corpus. Advantages of the approach are (i) fast tagger development time without linguistic engineering, (ii) accuracy better than or comparable to state of the art statistical and rule-based approaches, (iii) fast tagging speed, and (iv) reliable unknown word category guessing without the overhead of morphological analysis.

For a complete description of the approach and its application to tagging the American English Wall Street Journal corpus, see Daelemans et al. (1996). Before advancing to the results for tagging Dutch text, we will summarize the main characteristics of the approach.

2 Memory-Based Part of Speech Tagging

In order to make the problem amenable to Memory-Based Learning, the mapping from sentences to series of tags is approximated by a function from a set of features, representing the focus word and its fixed-width context, to the disambiguated tag belonging to the focus word. By doing this, the mapping becomes a *classification* task (Table 1) and we can apply Memory-based classifiers.

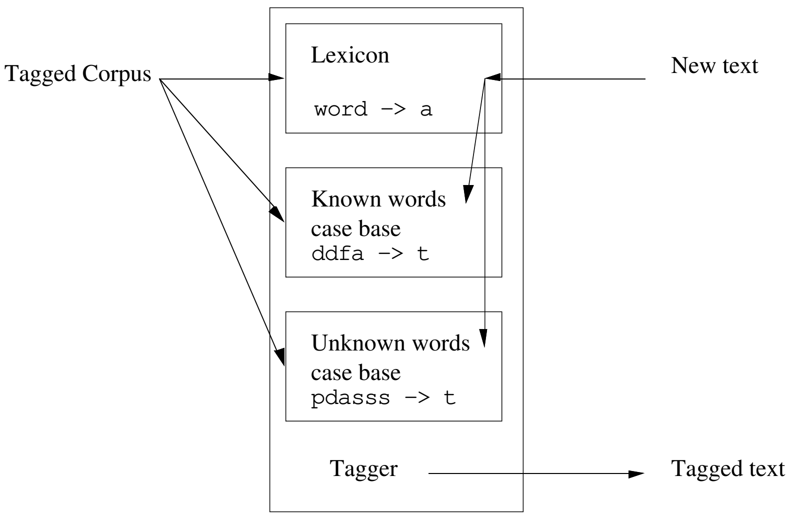
In the memory-based approach, a set of example cases is kept in memory. Each case consists of a word (or a lexical representation for the word) with preceding and following context, and the corresponding category for that word in that context. A new sentence is tagged by selecting for each word in the sentence the most similar case(s) in memory, and extrapolating the category of the word from these 'nearest neighbours'. The similarity metric used, considers the number of matching features between cases, and weighs the relative importance of each feature by an Information Gain factor. This number measures the utility of the feature in predicting the correct classification.

Table 1: Tagging as a classification task.

Input					Output
Left context		Focus	Right context		Category
=	=	John	will	join	np
=	John	will	join	the	md
John	will	join	the	board	vb
will	join	the	board	=	dt
join	the	board	=	=	nn

The architecture of MBT, our memory-based learning tagger (Figure 1 takes the form of a *tagger generator*: given a corpus tagged with the desired tag set, a POS tagger is generated which maps the words of new (untagged) text to tags in this tag set according to the same systematicity.

Figure 1: Architecture of the tagger-generator: flow of control.



The construction of a POS tagger for a specific corpus is achieved in the following way. Given an annotated corpus, three datastructures are automatically extracted: a *lexicon* (associating words to possible tags as evidenced in the training corpus), a case base for *known words* (words occurring in the lexicon), and a case base for *unknown words*. Case Bases are compressed using IGTree (Daelemans et al., 1997) for efficiency. During tagging, each word in the text to be tagged is looked up in the lexicon. If it is found, its lexical representation is retrieved and its context is determined, and the resulting pattern is disambiguated using extrapolation from the most similar cases in the known words case base. When a word is not found in the lexicon, its lexical representation is computed on the basis of its form, its context is determined, and the resulting pattern is disambiguated using extrapolation from the most similar cases in the unknown words case base. In each case, output is a best guess of the category for the word in its current context. We will describe each stage in the tagger construction process in some more detail.

2.1 Lexicon Construction

A lexicon is extracted by computing for each word in the training corpus the number of times it occurs with each category. E.g. when using the first 2 million words of the Wall Street Journal corpus[1] as training corpus, the word *once* would get the lexical definition *RB: 330; IN: 77*, i.e. *once* was tagged 330 times as an adverb, and 77 times as a preposition/subordinating conjunction.

Using these lexical definitions, a new, possibly ambiguous, tag is produced for each word type. E.g. *once* would get a new tag, representing the category of words which can be both adverbs and prepositions/conjunctions (RB-IN). Frequency order is taken into account in this process: if there would be words which, like *once*, can be RB or IN, but more frequently IN than RB (e.g. the word *below*), then a different tag (IN-RB) is assigned to these words.

2.2 Case Bases

For known words, cases consist of information about a focus word to be tagged, its left and right context, and an associated category (tag) valid for the focus word in that context. For unknown words, a tag can be guessed only on the basis of the *form* or the *context* of the word. In our memory-based learning approach, we provide word form information (especially about suffixes) indirectly to the tagger by encoding the three last letters of the word as separate features in the case representation. The first letter is encoded as well because it contains information about prefix and capitalization of the word. Context information is added to the case representation in a similar way as with known words.

In most taggers, some form of morphological analysis is performed on unknown words, in an attempt to relate the unknown word to a possible combination of known morphemes, thereby allowing its association with one or more possible categories. After determining this ambiguous category, the word is disambiguated using context knowledge, the same way as known words. Morphological analysis presupposes the availability of language-specific resources such as a morpheme lexicon, spelling rules, morphological rules, and heuristics to prioritise possible analyses of a word according to their plausibility. This is a serious knowledge engineering bottleneck when the goal is to develop a language and annotation-independent tagger generator.

Table 2 is a sample of the case base for the first sentence of the WSJ corpus (*Pierre Vinken, 61 years old, will join the board as a nonexecutive director nov. 29*) when using this case representation. The final column shows the target category; the disambiguated tag for the focus word. We will refer to this case representation as `ddfat` (d for disambiguated, f for focus, a for ambiguous, and t for target).

[1] ACL Data Collection Initiative CD-ROM 1, September 1991.

Table 2: Case representation for known words.

Word	Case representation				
	d	d	f	a	t
Pierre	=	=	np	np	np
Vinken	=	np	np	,	np
,	np	np	,	cd	,
61	np	,	cd	nns	cd
years	,	cd	nns	jj-np	nns
old	cd	nns	jj-np	,	jj
,	nns	jj	,	md	,
will	jj	,	md	vb	md
join	,	md	vb	dt	vb
the	md	vb	dt	nn-np	dt
board	vb	dt	nn-np	in-rb	nn
as	dt	nn	in-rb	dt	in
a	nn	in	dt	jj	dt
nonexecutive	in	dt	jj	nn-np	jj
director	dt	jj	nn-np	np	nn
nov.	jj	nn	np	cd	np
29	nn	np	cd	.	cd
.	np	cd	.	=	.

An interesting property of memory-based learning is that case representations can be easily extended with different sources of information if available (e.g. feedback from a parser in which the tagger operates, semantic types, the words themselves, lexical representations of words obtained from a different source than the corpus, etc.).

Table 3 shows part of the case base for unknown words. We will call this case representation `pdassst` (p for prefix letter, d for disambiguated category, a for ambiguous category, s for suffix letter, t for target category). As the chance of an unknown word being a function word is small, and cases representing function words may interfere with correct classification of open-class words, only open-class words are used during construction of the unknown words case base.

Table 3: Case representation for unknown words.

Word	Case representation						
	p	d	a	s	s	s	t
Pierre	P	=	np	r	r	e	np
Vinken	V	np	,	k	e	n	np
61	6	,	nns	=	6	1	cd
years	y	cd	jj-np	a	r	s	nns
old	o	nns	,	o	l	d	jj
join	j	md	dt	o	i	n	vb
board	b	dt	in-rb	a	r	d	nn
nonexecutive	n	dt	nn-np	i	v	e	jj
director	d	jj	np	t	o	r	nn
nov.	n	nn	cd	o	v	.	np
29	2	np	.	=	2	9	cd

2.3 Results on Wall Street Journal Corpus

For evaluation, we performed the complete tagger generation process on a 2 million words training set (lexicon construction and known and unknown words case-base construction), and tested on 200,000 test words. Generalization performance on known words (96.7%), unknown words (90.6%), and total (96.4%) is competitive with alternative rule-based and statistical approaches on the same corpus, and both training and testing speed are excellent (text tagging is possible with a speed of 1200 words per second). In contrast to statistical approaches, such as Hidden Markov Models, our approach does not need to estimate any parameters on the basis of the training data, so that the training corpora can be relatively small. For the WSJ tagger, we have found that a training corpus of around a hundred thousand words already gives very good performance.

3 Experimental Results for Dutch

We applied the MBT tagger-generator architecture to the written part of the Eindhoven corpus (Uit Den Boogaart), tagged using the WOTAN tagset developed by the TOSCA group of the Language and Speech department of the University of Nijmegen (Berghmans, 1995). In the experiment, the tagger was generated on the basis of the 610806 first words of the tagged example corpus (27651 sentences). The performance of the resulting tagger was tested on the 100,000 last words (5763 sentences) of the 710806 word Eindhoven corpus. Note that the tagger was therefore tested on a different sub-corpus of the corpus than it was trained on, which may have influenced accuracy negatively.

In the experiment, we restricted the tag set to the twelve main categories of the WOTAN tag set: N, V, Punc, Prep, Pron, Art, Adv,Adj, Conj, Num, Misc, Int. We added an additional tag "." as a more specific Punctuation tag for practical reasons (the tagger then knows when the sentence ends). This introduces an additional tag, bringing the total to 13 tags. This tag set is comparable to the one used by INL[2]. The results on known words, unknown words and overall performance is listed in Table 4.

Table 4: Tagging accuracy on known and unknown words.

	Accuracy	Percentage
Known	97.1	94.5
Unknown	71.6	5.5
Total	95.7	100.0

These results seem to be as good or better than those of state-of-the art rule-based and statistical approaches to tagging for Dutch. Tagging speed is fast (1280 word tokens per second). The most impressive aspect of this experiment is without any doubt the fast development time for the tagger (1 person-day for the complete process of corpus pre-processing, tagger generation, and testing). By adding further features for unknown words (such as `has-hyphen`, `contains-number` and `contains-uppercase`), it should be possible to get the unknown words in the same accuracy range as in our WSJ experiments.

For comparison, The TOSCA group (Nijmegen) reports a slightly lower performance for the WOTAN tagger on the Eindhoven written corpus (Van Halteren, p.c.). INL reports a tagging speed for DUTCHTALE of 87 tokens per second, and tagging accuracy between 89.5% and 91.5% correct on newspaper text.

4 Conclusion

We have applied the memory-based tagger generator to the construction of a tagger for English and for Dutch. We have shown that a memory-based approach to large-scale tagger construction is feasible both in terms of accuracy (comparable to or better than other statistical approaches), and also in terms of computational efficiency (time and space requirements). The approach combines some of the best features of learned rule-based and statistical systems (small training corpora needed, incremental learning, understandable and explainable behavior of the system). The most prominent advantages of the approach are the fast development time for constructing efficient high-accuracy taggers and the complete lack of linguistic engineering in terms of language specific tagging rules or morphological analyzers. We are currently working on the application of the described tagger for Dutch in a large scale document analysis environment.

[2] Institute for Dutch Lexicography

5 References

Berghmans, J. (1995). *WOTAN - een probabilistische grammatikale tagger voor het Nederlands.* Doctoraalscriptie, K.U. Nijmegen.

Brill, E. (1992). 'A simple rule-based part-of-speech tagger'. *Proceedings Third ACL Applied,* Trento, Italy, 152--155.

Church, K. (1988). 'A stochastic parts program and noun phrase parser for unrestricted text'. *Proceedings Second ACL Applied NLP*, Austin, Texas, 136--143.

Cutting, D., Kupiec, J., Pederson, J., Sibun, P. (1992). A practical part of speech tagger. *Proceedings Third ACL Applied NLP*, Trento, Italy, 133--140.

Daelemans, W., J. Zavrel, P. Berck, and S. Gillis. (1996). 'MBT: A Memory-Based Part of speech Tagger Generator.' In *Proceedings of the Fourth Workshop on Very Large Corpora*, Copenhagen: ACL SIGDAT, 14-27.

Daelemans, W., Van den Bosch, A., & Weijters, A. 'IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms.' To appear in *Artificial Intelligence Review*, special issue on Lazy Learning, 1997.

Karlsson, F., Voutilainen, A., Heikkila, J., Anttila, A. (1995). *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter: Berlin and New York.

van der Voort van der Kleij, J., Raaijmakers S., Panhuysen M., Meijering M., van Sterkenburg R. (1994). 'Een automatisch geanalyseerd corpus hedendaags Nederlands in een flexibel retrievalsysteem.' In: Noordman, L. & W. Vroomen (red.) *Informatiewetenschap 1994*. Tilburg: STINFON, 181-194, 1994.