Lemmatisation and Morphosyntactic Annotation for the Spoken Dutch Corpus

Frank Van Eynde
Center for Computational Linguistics
Maria-Theresiastraat 21
3000 Leuven, Belgium
frank.vaneynde@ccl.kuleuven.ac.be

Jakub Zavrel and **Walter Daelemans**
CNTS / Language Technology Group
Universiteitsplein 1
2610 Wilrijk, Belgium
{zavrel,daelem}@uia.ua.ac.be

Monachesi (ed.), Proceedings 10th CLIN meeting, p. 53-62, 2000

Abstract

This paper describes the lemmatisation and tagging guidelines developed for the "Spoken Dutch Corpus", and lays out the philosophy behind the high granularity tagset that was designed for the project. To bootstrap the annotation of large quantities of material (10 million words) with this new tagset we tested several existing taggers and tagger generators on initial samples of the corpus. The results show that the most effective method, when trained on the small samples, is a high quality implementation of a Hidden Markov Model tagger generator. However, we also show that a combination of systems improves the accuracy, and that this combinatory approach allows us to leverage existing taggers with different tagsets and lexical resources.

1 Introduction

The Dutch-Flemish project "Corpus Gesproken Nederlands" (1998-2003) aims at the collection, transcription and annotation of ten million words of spoken Dutch, see Oostdijk (this volume). The first layer of linguistic annotation concerns the assignment of base forms and morphosyntactic tags to each of those ten million words. This paper presents the lemmatisation guidelines (Section 2) and the tagset (Section 3) which have been devised for this purpose. It also describes the evaluation procedure which has been adopted for the selection of a lemmatiser and a tagger (Section 4 and Section 5), and it provides the results of an experiment in which the results of different taggers are combined (Section 6).

2 Lemmatisation

Each of the ten million word forms which occur in the corpus has to be paired with a corresponding base form (lemma). For verbs, this base form is identified with the infinitive, and for most other words with the stem, i.e. a form without inflectional affixes. The noun *stoelen* (chair + PLURAL), for instance, is paired with *stoel*, the adjective *mooie* (beautiful + DECLENSION) with *mooi*, and the numeral *vijfde* (five + ORDINAL) with *vijf*. Truncated forms, on the other hand, are paired with the corresponding full forms; the article in *'t station* (the station), for instance, is paired with *het*, the possessive in *z'n hond* (his dog) with *zijn*, and the pronoun in *kent 'm* (knows him) with *hem*.[1] In many cases, the base form is identical with the word form itself, cf. the conjunctions, prepositions and interjections, plus the uninflected forms of nouns, adjectives and numerals.

The pairing with base forms, as it is performed in CGN, is subject to three general constraints. First, the base form must be an independently existing word form. A plurale tantum, such as *hersenen* (brains), for instance, is not paired with *hersen*, but rather with *hersenen*. By the same token, the base form of an inherently diminutive noun like *meisje* (girl) is not identified with *meis* or *meid*, but rather with *meisje* itself. Idem dito for the genitive pronoun in *andermans zaken* (someone else's business) and the comparative in *eerdere pogingen* (previous attempts). Second, the pairing with base forms is performed on a word-by-word basis. The individual word forms in a sentence like

(1) Hij belt haar elke dag op.
He rings her every day up.
'he calls her every day'

[1] Notice that the base form is *hem*, rather than *hij*, since the distinction between nominative and oblique pronouns is not made in terms of inflectional affixes.

are paired with resp. *hij, bellen, haar, elk, dag* and *op*. That *belt* and *op* are part of the discontinuous verb *opbellen* (call) is not recognised at this level, since it would require a full-fledged syntactic analysis of the clause. Third, each word form must receive one and only one base form. In

(2) Daar vliegen van die rode vliegen.
There fly of those red flies.
'there are some red flies flying over there'

the first occurrence of *vliegen* must be paired with the infinitive *vliegen* (to fly), whereas the second occurrence must be paired with the noun stem *vlieg* (a fly). For a systematic disambiguation of this kind, the lemmatiser needs access to part-of-speech information, which is the topic of the next section.

3 Tagset

The tags which are assigned to the word form tokens consist of a part-of-speech value and a list of associated morpho-syntactic features. The content of the tags is specified by the tagset. This section first presents the requirements which we want the tagset to fulfill (3.1), and then provides a formal definition of the tagset (3.2). Special attention is paid to the selection of the morpho-syntactic features (3.3) and to the context-dependent assignment of the tags (3.4).

3.1 Evaluation criteria

Many of the tagsets which are currently used for the analysis of Dutch have a rather low level of granularity: the number of tags which they employ typically ranges from 10 to 50 (INL 11, KEPER 24, D-TALE 45, XEROX 49).[2] For many applications, this may be sufficient, but for CGN we are aiming at a higher level of granularity, since the tags will be the only form of linguistic annotation for 90 % of the corpus (the second layer of annotation, syntactic analysis, will cover only 10 % of the corpus). A second requirement for the CGN tagset is modularity: in many tagsets each tag is treated as an atom, and while this practice may be appropriate for systems with a low level of granularity, it leads to a high degree of redundancy in systems with a high level of granularity. As a consequence, we will not work with monadic tags like VAUXFINPL, but rather with structured tags like V(aux,finite,plural). This modularity is not only an asset in itself, it also facilitates further syntactic processing, since the different pieces of information in the tag may serve different roles and functions in the syntactic representation. A third requirement concerns the content of the tags. Since the annotation should be accessible and useful for a broad spectrum of potential users, the tagset should draw as much as possible on sources which are commonly available and relatively familiar. For Dutch, the prime source in this respect is the Algemene Nederlandse Spraakkunst (Haeseryn et al., 1997). A fourth requirement concerns the existence of extensive and easily accessible documentation: a mere listing of the tags accompanied with some examples may be sufficient for systems which only distinguish ten to twenty different tags, but for a system with high granularity the absence of documentation would seriously compromise the value of the annotation. A fifth requirement, finally, concerns the conformity to international standards. Especially in multi-lingual Europe, there have been various initiatives in the nineties aiming at cross-lingual standards or guidelines for linguistic analysis. The most influential in the field of POS tagging are EAGLES (1996) and MULTEXT (1996).

As of 1998, when the CGN project started, there were two Dutch tagsets which came close to meeting most of these requirements, i.e. WOTAN (1 and 2) and PAROLE. However, WOTAN-1 was being phased out and WOTAN-2, which was to replace it, was not stable: it kept changing during the preparatory phase of the CGN project and was still 'under construction' in April 1999 (Van Halteren, 1999). The problem with PAROLE, on the other hand, was

[2] The INL tagset is named after the *Instituut voor Nederlandse Lexicologie* (Leiden); it is used a.o. in CORRie, a system for automatic spelling correction, developed by Theo Vosse at Leiden University, see Vosse (1994). The KEPER tagset was developed by Polderland BV and is used in information retrieval applications. The D-TALE tagset was developed by the Lexicology Group at the Vrije Universiteit Amsterdam and is mainly used for lexicographic purposes. The XEROX tagset was developed at the Xerox laboratories in Grenoble; it is the Dutch counterpart of a French original, described in Chanod and Tapanainen (1995).

the scarcity of documentation. For this reason, it was decided to design a new tagset for CGN, taking into account the five requirements above.

3.2 Formal definition

Formally, the CGN tagset is a six-tuple $< A, V, P, D, I, T >$, where A is a set of attributes, V of values, P of partitions, D of declarations, I of implications, and T of tags. Features are pairs of attributes and values, such as '`NUMBER = plural`'.[3] The values we use will all be atomic, i.e. they do not consist of an attribute-value pair in turn. Partitions specify for each attribute what its possible values are, as in

[P] `NUMBER = singular, plural`.

Features are combined into lists, such as <`NUMBER = plural, GENDER = neuter`>. A subset of the possible combinations correspond to tags. This subset is singled out by declarations and implications. The former specify which attributes are appropriate for which tags, as in

[D] <`POS = noun`> ⇒ <`NUMBER, DEGREE, GENDER`>

The latter specify dependencies between the values of different features in the same tag, as in

[I] <`DEGREE = diminutive`> ⇒ <`GENDER = neuter`>

This one specifies that (Dutch) diminutives have neuter gender. Tags are lists of features, which satisfy all of the declarations and implications. An example is

[T] <`POS = noun, NUMBER = singular, DEGREE = diminutive, GENDER = neuter`>

For reasons of brevity, this full format is reduced to mnemonic tags like `N(sing,dim,neuter)`.[4]

[3]The examples in this paragraph are only meant for illustration. In the next paragraph we provide some examples form the tagset itself.

[4]To emphasise the language specific nature of the features, the CGN tagset makes use of Dutch names for both the attributes and the values. The use of English names in this paper is just for expository purposes.

3.3 Selection of the features

The tokens which are the basic units of the orthographic representation come in three types.

[P01] `TOKEN = word, special, punctuation`.

The words are associated with a part-of-speech attribute, whose values are listed in [P02].

[D01] <`TOKEN = word`> ⇒ <`POS`>

[P02] `POS = noun, adjective, verb, pronoun, article, numeral, preposition, adverb, conjunction, interjection`.

These ten values correspond one-to-one to the parts-of-speech which are distinguished in the Algemene Nederlandse Spraakkunst (Haeseryn et al., 1997). The special tokens do not receive a `POS` value, but a `SPECTYPE` feature which specifies whether the token is foreign, incomplete, incomprehensible[5] or exceptional in some other way.

[D02] <`TOKEN = special`> ⇒ <`SPECTYPE`>

[P03] `SPECTYPE = foreign, incomplete, incomprehensible, ...`.

The punctuation signs, finally, do not receive any extra features. According to EAGLES (1996), these are the distinctions which a tagset should minimally include.

As we are aiming for high granularity, though, there are various other features which need to be added. More specifically, we will add features for

- distinctions which are marked by inflection, such as `NUMBER` for nouns and `MOOD/TENSE` for verbs, or by highly productive category preserving derivation, such as `DEGREE` for nouns;
- distinctions which reflect lexical properties of the word form (as opposed to the base form), such as `GENDER` for nouns; notice, for instance, that the diminutive *stoeltje*

[5]Remember that CGN is a corpus of *spoken* Dutch.

is neuter, whereas the corresponding base form *stoel* is non-neuter;

- a number of commonly made morpho-syntactic distinctions, such as `CONJTYPE` for conjunctions (`coordinating` vs. `subordinating`), and `NTYPE` for nouns (`proper` vs. `common`).

By way of example, we mention the relevant declarations and partitions for the nouns.

[D03] <`POS = noun`> ⇒ <`NTYPE, NUMBER, DEGREE`>
[D04] <`POS = noun, NUMBER = singular`> ⇒ <`CASE`>
[D05] <`POS = noun, NUMBER = singular, CASE = standard`> ⇒ <`GENDER`>

[P04] `NTYPE = common, proper`.
[P05] `NUMBER = singular, plural`.
[P06] `DEGREE = base, diminutive`.
[P07] `CASE = standard, genitive, dative`.
[P08] `GENDER = neuter, non-neuter`.

All nouns are marked for `NTYPE`, `NUMBER` and `DEGREE`, but `CASE` is only assigned to the singular nouns, since the distinction is systematically neutralised in plural nouns, and `GENDER` is only assigned to the singular standard nouns, since it is neutralised in the plural, the genitive and the dative. In this way, we ensure that features are only assigned when the distinctions which they make are relevant; there is, hence, no need for values like *non-applicable*.

Given these declarations and partitions, the number of nominal tags amounts to twenty, but four of those are ruled out by the implications

[I01] <`POS = noun, NUMBER = singular, DEGREE = diminutive`> ⇒ <`CASE` ≠ `dative`>
[I02] <`POS = noun, NUMBER = singular, DEGREE = diminutive, CASE = standard`> ⇒ <`GENDER = neuter`>

In words, the diminutive nouns are never dative and always neuter.

For each of the ten parts-of-speech, the tagset contains the relevant declarations, partitions and implications, see Van Eynde (2000). It is not possible to present them all in this paper, but Figure 1 gives a full survey of the relevant declarations.

All in all there are 25 declarations, 24 partitions and 313 tags; almost two thirds of them belong to the pronoun/determiner part-of-speech.

Not included in the tagset are features for semantic distinctions. The fact that the noun *vorst*, for instance, is ambiguous between a kind of ruler (sovereign) and a kind of weather (frost), is not made explicit in the tagset.

3.4 The assignment of tags

3.4.1 Form vs. function

As for the assignment of tags to the individual tokens, CGN follows the principle that (morpho-syntactic) form prevails over (syntactic) function and meaning. To illustrate, let us take the number distinction for nouns. In the following sentences the NPs just after the verb denote an aggregate of resp. tourists and prisoners.

(3) Er komt een groep toeristen aan.
There comes a group tourists on.
'a group of tourists arrives'

(4) Er zijn een aantal gevangenen ontsnapt.
There are a number prisoners escaped.
'a number of prisoners escaped'

In spite of this semantic plurality, though, both *groep* (group) and *aantal* (number) are treated as singular, since they lack the affixes which are typical of plural nouns.

3.4.2 Disambiguation

To make the tagged corpus as informative as possible, we are aiming at complete disambiguation. This implies that each word form token should be assigned exactly one tag, more specifically the one that is appropriate in the given context. This is of course harder to achieve for a tagset with high granularity than for a coarse-grained one, and since CGN definitely belongs to the former, it is important to design it in such a way that it does not create an insurmountable amount of ambiguity.

To demonstrate what is meant by this, let us make a distinction between occasional and systematic ambiguity. An example of the former is the POS-ambiguity of the word *bij*, which can

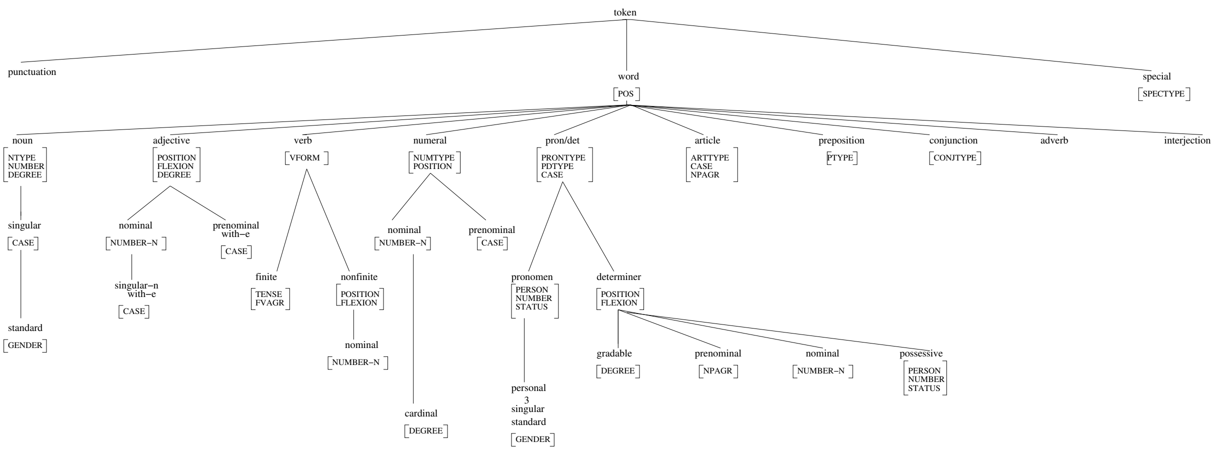

Figure 1: An overview of the tagset definition hierarchy.

either be a noun (bee) or a preposition (with), or of the word *arm*, which can either be a noun (arm) or an adjective (poor). Such ambiguities have to be taken as they are, and should also be resolved. Systematic ambiguities, however, can —to some extent— be avoided. Many of the Dutch prepositions, for instance, are not only used to introduce an NP or some other complement, but can also be used without adjacent complement, as a consequence of stranding or intransitive use. Compare, for instance, the different uses of *boven* (above) in

(5) Niemand staat boven de wet.
Nobody stands above the law.
'nobody is above the law'

(6) Daar gaat niets boven.
There goes nothing above.
'that's the best there is'

(7) Ze zijn boven.
They are above.
'they are upstairs'

(8) Olie drijft altijd boven.
Oil floats always above.
'oil will always float'

Since most of the other prepositions show similar types of versatility, we would be left with a systematic POS-ambiguity, if we were to treat them as ambiguous between say 'preposition', 'adverb' and 'particle'. If, on the other hand, these are treated as different possible uses of prepositions, there is no ambiguity at the POS-level; the finer-grained distinctions are then left to syntactic analysis.

Another way to diminish the amount of systematic ambiguity is to allow for underspecification.

3.4.3 Underspecification

In paragraph 3.3 the `CASE` values have been identified as in

[P07] `CASE = standard, genitive, dative`.

For the personal pronouns, though, one should also make a distinction between nominative (*ik, jij, hij, we, ...*) and oblique (*mij, jou, hem, ons, ...*). At the same time, it would be inappropriate to apply this finer distinction to the nouns, since no Dutch noun has different forms for the nominative and the oblique. Rather than introducing a systematic ambiguity for all nouns we allow for some variation in the partition.

[P07] `CASE = standard (nominative, oblique), special (genitive, dative)`.

The basic `CASE` distinction is the one between 'standard' and 'special', corresponding resp. to forms without and with case suffix. The former can be further partitioned in nominative and oblique, and the latter in genitive and dative, but whether these finer-grained distinctions apply depends on the part of speech. For the pronouns they both do, but for the nouns it is only the latter which applies, and for the adjectives neither of the two.

Another example concerns gender. While the majority of nouns is either neuter or non-neuter, there are some which can be either. If the use of different genders corresponds with a clear semantic distinction, as in *de bal* (round object) vs. *het bal* (dancing occasion) or *de blik* (the look) vs. *het blik* (the can), we distinguish between a neuter gender noun and a non-neuter gender noun. If, however, the gender variation does not correspond to any clear semantic distinction, as in *de/het filter* (the filter) or *de/het soort* (the kind), we only assign a specific value when the noun is accompanied by a determiner which is overtly neuter or non-neuter, such as the definite article. In the absence of such a determiner, we allow the assignment of a generic value, as in

[P08] `GENDER = gender (neuter, non-neuter).`

While the allowance for underspecification is—in principle—an asset, it also has the potential disadvantage of increasing the number of possible tags and hence the amount of ambiguity. For the nouns, for instance, we had sixteen possible combinations of feature values (see 3.3), but with the allowance of underspecified gender we have to foresee two more: one for the common nouns and one for the proper nouns. For this reason, we have made a very modest use of underspecification.

3.4.4 The need for automatization

In principle, it would be possible to do the POS tagging manually. Given the 313 possible tags and the criteria for assigning them, as specified in Van Eynde (2000), it could be left to humans to assign a contextually appropriate tag to each individual token. However, given the size of the corpus (ten million words) and the scarcity of the resources which can be spent on the task, this option would be unrealistic. Moreover, given the state of the art in POS tagging, it is reasonable to expect that automatic taggers can deliver results which are accurate enough to be useful as a first draft. In order to find out which taggers or tagger generators give the most promising results for the CGN tagset we did some detailed comparative evaluation of the available tools. The results of this evaluation are summarized in the next section.

4 Selection of tagger and lemmatiser

This and the following sections describe the selection of an automatic tagger and lemmatiser for the (partially) automated annotation of the CGN corpus, using the tagset specified above. A more detailed account of the selection of the tagger is given in Zavrel and Daelemans (1999).

The selection of a lemmatiser was limited to four candidates: the system from XEROX, using finite state rules, the MBMA system using memory-based learning (Van den Bosch and Daelemans, 1999), the KEPER system, and the rule/lexicon-based D-Tale system. The results of a test of these systems on the initial corpus sample SMALL-1 (described below) is shown in Table 1. The main differences were due to the verbs, i.e. reduction to stem vs. reduction to the infinitive. After this was discounted, the results were, in general, satisfactory and a choice was made, on the basis of direct availability, to use MBMA.

Automatic morphosyntactic tagging normally presupposes a tagger that uses the appropriate tagset, or a tagger generator and a sufficiently large annotated corpus that can be used to train such a tagger. Both of these prerequisites were not available in our situation because of the newly designed tagset. Therefore, we examined available resources with two goals in mind: First, the need to bootstrap the initial part of the corpus. For this we might be able to use an existing tagger with a different tagset. In this case it is important that the tagger is accurate in terms of its own tagset, and that there is an easy mapping to the CGN tagset. Second, once enough data is correctly annotated, a tagger generator with high accuracy is needed to train taggers specifically adapted to both the CGN tagset (i.e. high granularity etc.), and the CGN annotation process (i.e. giving more than one choice, indicating certainty, and being easy to retrain).

The selection of a tagger considered two types of candidates: taggers only available with existing tagsets and tagger generators which were

	% error			
	MBMA	D-Tale	Xerox	KEPER
total	18.2	5.3	6.7	16.1
excluding verbs	3.6	3.6	5.8	4.8

Table 1: Error rate of lemmatisation on SMALL-1.

available trained on WOTAN 1 or 2 material from the Eindhoven corpus in the context of an earlier tagger comparison (Van Halteren et al., 1998). The first category consisted of the before mentioned XEROX, KEPER, and D-TALE systems, augmented with an HMM tagger from the CORRie system (Vosse, 1994). The second group of WOTAN trained tagger generators contained: MBT, a memory based tagger (Daelemans et al., 1996), MXPOST (Ratnaparkhi, 1996), Eric Brill's rule-based system (1994), and TnT, a state-of-the-art HMM implementation (Brants, 2000).

5 Experiments on CGN data

5.1 Data

For the experiments a small sample of transcripts from the initial CGN corpus was annotated manually by three independent annotators. After filtering out punctuation from the sample of some 3000 tokens, a total of only 2388 tokens were left for testing purposes. Because the tagset and guidelines were still under development at that moment, the inter-annotator agreement was quite low. Therefore a majority vote of the three annotators was taken as a benchmark for the following experiments. The (few) ties were resolved manually by consensus. We will refer to this data set as SMALL-1. For more details of the construction of this data set, see Zavrel (1999). Later, after more data was available, and the tagset had converged, several experiments were repeated with larger samples: BATCH-1 and BATCH-2, counting respectively 22786 and 39304 tokens (including punctuation). All accuracy measurements on train/test experiments given below were performed using tenfold cross-validation, except where noted otherwise.

5.2 Results

5.2.1 Native tagset

A first measurement concerns the accuracy of each existing tagger in terms of the distinctions that its own "native" tagset makes. Since, in general, no benchmark data sets are available in those tagsets, and no tagging manual is available for most tagsets, a rough accuracy estimate was made on the basis of the CGN benchmark data. For this purpose, only those tags that were in clear contradiction with the benchmark were counted as errors. E.g. a tag of Propernoun, where the benchmark says Adjective is clearly wrong, whereas a tag of Verb where the benchmark says V(finite,present,3sing) is counted as correct. Thus differences in granularity usually weigh in favor of the less fine-grained tagging. The results are shown in Table 2. The taggers with fixed tagsets are generally less accurate than the WOTAN based taggers, even though these have much larger tagsets; among them TnT is the best one.

5.2.2 Mapping to the CGN tagset

When we want to bootstrap from an existing tagset, it is not only important how accurate a given tagger handles that tagset (see previous section), but also how difficult it is to translate the correct tag in the source tagset to the correct tag in the CGN tagset. In this section, we set aside all issues of tagging style and guidelines, and estimate the complexity of this mapping in purely statistical terms. After we have collected all the tagger outputs on our test sample, we can measure the amount of uncertainty that is left about the correct CGN tag. For this we use the Information Gain measure, or its variant Gain Ratio (Quinlan, 1993). The latter is normalised with respect to the size of the source tagset. The corresponding numbers are summed up in Table 3. We also included the word to be tagged itself as if it were an existing source tag (first column). Again, the best values are found for the WOTAN-1 based taggers, among which TnT has the best behaviour. A more practical measure of the mapping difficulty is given by the number of correct tags that we get when we translate each source tag

tagger	D-Tale	KEPER	XEROX	CORRie	WOTAN-1 MBT	WOTAN-1 MX	WOTAN-1 TnT	WOTAN-2 MBT	WOTAN-2 MX	WOTAN-2 TnT
tagset size	45	24	49	11	347	347	347	1256	1256	1256
accuracy (%)	82.4	73.7	78.8	86.7	87.8	86.9	89.9	82.9	81.8	83.9

Table 2: Rough estimates of accuracy percentages in terms of each system's own tagset. (MX stands for the MXPOST tagger.)

to its most likely CGN translation. These figures are given on the bottom row of the table. This measure, which is not entirely unrealistic with regards to an automatic conversion between source and target, shows that the highly detailed WOTAN-2 tagset is at an advantage over its more coarse-grained competitors.

5.2.3 Training from scratch

The previous sections show that the chances of obtaining high accuracy taggings (90-95% correct) by using existing taggers and tagsets for the CGN material are not very good. This is a harmful situation for a quick bootstrap phase of the corpus annotation process. Typically, taggers are trained on data sets of tens or hundreds of thousands of tokens, and it is very laborious to annotate such quantities if approximately every fifth word needs to be manually corrected. So we wanted to see, as a calibration point, what the accuracy would be of the available tagger generators, trained on the minimal sample (SMALL-1) of available hand-tagged material. The figures in Table 4 show the average results from a ten fold cross-validation experiment. Again TnT turns out to be superior. It is interesting to see that on such a small training sample the accuracy is already higher than that obtained by mapping.

	MBT	MX	BRILL	TnT
%	80.6	69.7	78.2	82.7

Table 4: Accuracy percentages of the four tagger generators when trained and tested (ten fold cross validation) on the CGN tagset annotated sample SMALL-1.

6 Tagger Combination

Previous work (Van Halteren et al., 1998; Tufiş, 1999) has shown that a combination of systems often gives an accuracy better than that of the best system for Part-of-speech tagging. We have also explored this direction in the present context. Using the setup described in Van Halteren et al. (1998), it was first thought that one is confined to a combination of systems that produce outputs from the same tagset. The idea is that different systems, e.g. different because they are based on different learning algorithms or information sources, tend to produce different errors, and that a second level decision making process such as voting can eliminate minority errors. In more recent work (Van Halteren et al., 2000), we have further investigated the possibilities of *stacking*. In stacking, a second level machine learning algorithm is trained to produce the correct category (from any arbitrary tagset), using the outputs of the first level components (taggers) as features (using any arbitrary tagset). Using this method we are no longer confined to a combination of same-tagset taggers. We can integrate any number of existing taggers or, in fact, any type of lexical resource, to contribute to the quality of the tagging decisions.

Using TiMBL[6] (Daelemans et al., 2000), a Memory Based Learning system, as the combination method we have applied this approach with the available tagger outputs, and the word to be tagged, as a feature. On the SMALL-1 dataset, this has resulted in a best accuracy for the combined system of 86.6 %, compared with the best system, TnT, trained from scratch (82.7 %). In Table 5, an overview is shown of the contribution of the features to the combination. As the effect of combination is substantial, we have recommended the use of the combination strategy for bootstrapping the corpus, until a single tagger (TnT) reaches levels of accuracy that are comparable. Because not all taggers were as easily available as the WOTAN-based systems, we have restricted further experiments with combinations to these and the

[6] Available from `http://ilk.kub.nl/`

	word	D-Tale	KEPER	CORRie	WOTAN-1			WOTAN-2		
					MBT	MX	TnT	MBT	MX	TnT
IG (bits)	5.21	3.74	2.96	3.00	4.43	4.50	4.60	4.59	4.74	4.79
GR	0.66	0.82	0.80	0.84	0.82	0.84	0.86	0.76	0.77	0.77
accuracy (%)	70.2	50.6	42.8	42.6	71.6	72.6	75.9	72.7	77.2	77.5

Table 3: Information Gain and Gain Ratio of each tagger with respect to the desired target tagset. The bottom line gives an estimate of the accuracy after mapping each system's native tag to the most likely CGN tag.

CGN systems trained from scratch. More recent experiments on somewhat larger quantities of more consistent training material BATCH-1 and BATCH-2 show that this produces quite good results (see Table 6). This table also shows that the combination method still outperforms the single best tagger by a considerable margin on larger data sets. A more detailed discussion of bootstrapping through combination is provided in Zavrel and Daelemans (2000).

7 Conclusion

For tagsets with a low degree of granularity it is often not necessary to invest a lot of effort in precise definitions and documentation: most of the distinctions speak, as it were, for themselves. Likewise, the construction or the training of automatic taggers for such tagsets is relatively straightforward, since it can be based on comparatively small amounts of rules and/or data. Tagsets with a high degree of granularity, however, such as the one of CGN, are much more demanding, both conceptually and computationally. Conceptually, they require more extensive documentation and more precise guidelines for the assignment of tags to individual tokens. Computationally, they require considerably more rules (for a rule based tagger) or larger training samples (for a probability based tagger or a tagger generator) to arrive at useful results. The purpose of this paper was to show how these problems are dealt with in the framework of the CGN project. More specifically, we have presented the lemmatization guidelines and the high granularity CGN tagset, and we have described the selection of an automatic tagger for the bootstrapping of the corpus annotation process with this newly designed tagset. This led to the choice of the MBMA lemmatizer and the tagger generator TnT. However, while the latter is our system of choice in the medium and long term, it should be added that for the initial stages of the bootstrapping process, the best results were obtained when reusing many exisiting and newly trained taggers in a system combination framework.

Acknowledgements

For their comments and suggestions we would like to thank Antal van den Bosch, Hans van Halteren, Richard Piepenbrock, Ineke Schuurman, Lisanne Teunissen and the audiences at the first CGN users workshop (Tilburg, August 1999), the first international CGN workshop (Tilburg, November 1999), the tenth CLIN conference (Utrecht, December 1999) and the second LREC conference (Athens, June 2000).

References

T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000, April 29 – May 3, 2000, Seattle, WA.*

E. Brill. 1994. Some advances in transformation-based part-of-speech tagging. In *AAAI'94*.

J-P. Chanod and P. Tapanainen. 1995. Tagging French – comparing a statistical and a constraint-based method. In *Proceedings of EACL-95, Dublin.*

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In E. Ejerhed and I.Dagan, editors, *Proc. of Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2000. TiMBL: Tilburg memory based learner, version 3.0, reference manual, technical report ILK-0001. Technical report, ILK, Tilburg University.

all	word	D-Tale	KEPER	CORRie	WOTAN-1			WOTAN-2		
					MBT	MX	TnT	MBT	MX	TnT
86.3	85.9	86.6	86.4	86.1	86.0	85.9	86.1	86.4	86.3	86.5

Table 5: System combination results. The "all" column gives the score of the ensemble with all taggers and the word to be tagged as components. The remaining columns give the score of the combiner when the tagger in that column is left out. These scores are obtained from a test on a single test set of 10% of the total material.

dataset	MBT	MX	BRILL	TnT	EXISTING-combi	CGN+WOTAN-combi
SMALL-1	80.6	69.7	78.2	82.7	86.3	–
BATCH-1	89.4	89.4	86.3	91.6	–	94.3
BATCH-2	91.2	90.1	87.9	92.7	–	94.3

Table 6: System combination results on progressively cleaner and larger training sets. SMALL-1 was not very consistent and only 2985 tokens. BATCH-1 and BATCH-2 seem to be much 'cleaner' data, and contain respectively 22786 and 39304 tokens.

EAGLES. 1996. Recommendations for the morphosyntactic annotation of corpora. Technical report, Expert Advisory Group on Language Engineering Standards, EAGLES Document EAG - TCWG - MAC/R, March 1996.

W. Haeseryn, K. Romijn, G. Geerts, J. de Rooij, and M.C. van den Toorn. 1997. *Algemene Nederlandse Spraakkunst*. Martinus Nijhoff, Groningen & Wolters Plantyn, Deurne, tweede, geheel herziene druk edition.

MULTEXT. 1996. Lexical specifications. document LEX1. version 0.1. Technical report, December.

J.R. Quinlan. 1993. C4.5*: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proc. of the Conference on Empirical Methods in Natural Language Processing, May 17-18, 1996, University of Pennsylvania.*

D. Tufiş. 1999. Tiered tagging and Combined Language Models Classifiers. In *Proceedings Workshop on Text, Speech, and Dialogue, Brno.*

A. Van den Bosch and W. Daelemans. 1999. Memory-based morphological analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland, USA*, pages 285–292.

F. Van Eynde. 2000. Part of speech tagging en lemmatisering. Technical report, CGN-Corpusannotatie. Working Paper, May.

H. Van Halteren, J. Zavrel, and W. Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *Proceedings of ACL-COLING'98, Montreal, Canada*, pages 491–497.

H. Van Halteren, J. Zavrel, and W. Daelemans. 2000. Improving accuracy in NLP through combination of machine learning systems. *Submitted.*

H. Van Halteren, 1999. *The WOTAN2 Tagset Manual (under construction).* Katholieke Universiteit Nijmegen, April.

T. Vosse. 1994. *The Word Connection. Grammar-based Spelling Error Correction in Dutch.* Neslia Paniculata, Enschede.

J. Zavrel and W. Daelemans. 1999. Evaluatie van part-of-speech taggers voor het Corpus Gesproken Nederlands. Technical report, CGN-Corpusannotatie. Working Paper, July.

J. Zavrel and W. Daelemans. 2000. Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of LREC-2000.*

J. Zavrel. 1999. Annotator-overeenstemming bij het manuele taggingexperiment. Technical report, CGN-Corpusannotatie. Working Paper, June.