# Part of Speech Tagging and Lemmatisation for the Spoken Dutch Corpus

**Frank Van Eynde***, **Jakub Zavrel**†, **Walter Daelemans**†

*Center for Computational Linguistics
Maria-Theresiastraat 21
3000 Leuven, Belgium
frank.vaneynde@ccl.kuleuven.ac.be
†CNTS / Language Technology Group
Universiteitsplein 1
2610 Wilrijk, Belgium
{zavrel,daelem}@uia.ua.ac.be

**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.

LREC-2000, Athens, Greece, 1427-1434, 2000

## 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 (Oostdijk, 2000). The first layer of linguistic annotation concerns the assignment of base forms and morphosyntactic tags to each of those ten million words. The first part of this paper presents the lemmatisation guidelines and the tagset which have been devised for this purpose (Sections 2. and 3.). The second part focuses on the evaluation procedure which has been followed for the selection of a lemmatiser and a tagger (Sections 4. and 5.).

## 2. Lemmatisation

Each of the ten millions word forms which occur in the corpus is 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, e.g. 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*, but rather with *meisje* itself. Second, the pairing with base forms is performed on a word-by-word basis. In

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

the individual word forms 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 rare witte vliegen.
There fly of those strange white flies.
'there are some strange white 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 obviously 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).

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

### 3.1. Evaluation criteria for the CGN tagset

Many of the tagsets which have been made 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 (e.g. for the taggers that will be discussed in Section 4.: INL/CORRie11, KEPER 24, D-TALE 45, XEROX 49). 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 linguistic form of annotation for 90% of the corpus (the second layer of annotation, syntactic analysis, will cover only 10% of the corpus). A second requirement which the CGN tagset ought to meet 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. In order to guarantee high quality the output of the automatic tagging process is manually screened and —if necessary— corrected; the results of this screening will obviously be more reliable and uniform if the correctors can base their choices on commonly accepted and clearly defined guidelines. 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 is (EAGLES, 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 yet finished: it kept changing during the preparatory phase of CGN and was still 'under construction' in April 1999 (Van Halteren, 1999). A similar remark applies to PAROLE, which was announced to replace the older Dutchtale, but which was not available in 1998, and about which even now (March 2000) hardly any documentation can be found. For this reason, it was decided to design a new tagset for CGN, taking into account the five requirements above.

### 3.2. Formal definition of the tagset

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`'; 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[2]

[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)`.[3]

### 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,

[2]The examples in this paragraph are only meant for illustration. In the next paragraph we provide some real examples.

[3]In order to stress 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.

incomplete or incomprehensible.

[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; compare *het stoeltje* vs. *de stoel*;
- 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.

Given these declarations and partitions, the number of different 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 (singular) 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 app. 300 maximally specific tags (16 for the nouns, 30 for the adjectives, 26 for the verbs, 2 for the conjunctions, etc.); by far the largest class of tags belongs 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. What is also not included are valency distinctions. The reason for this omission can be illustrated with the verb *lachen* (laugh). Just like its English equivalent, *lachen* is intransitive, i.e. it cannot take an NP object.

(3) * Hij lacht ons.
* He laughs us.

When it is part of a discontinuous verb, though, as in the combination with *uit* (out), *toe* (to) or *weg* (away), it is strictly transitive, in the sense that it requires an NP object.

(4) Hij lacht ons uit.
He laughs us out.
'he is making fun of us'

(5) De toekomst lacht ons toe.
The future laughs us to.
'our future is looking bright'

(6) Ze lacht hun bezwaren weg.
She laughs their objections away.
'she laughs away their objections'

This demonstrates that valency patterns had better be assigned to the discontinuous verb as a whole, and, hence, at a level of analysis which goes beyond the word-by-word approach of the tagging.

### 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 finite verb denote an aggregate of resp. tourists and prisoners

(7) Er is een groep toeristen aangekomen.
There is a group tourists arrived.
'a group of tourists arrived'

(8) 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.

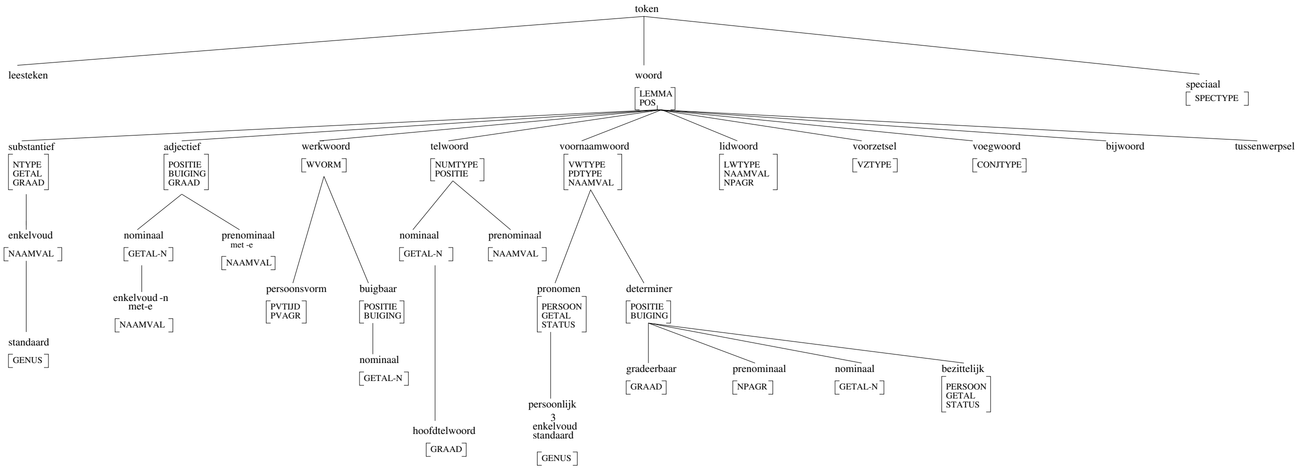


Figure 1: An overview of the tagset definition hierarchy.

### 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 first make a distinction between occasional and systematic ambiguity. An example of the former is the POS-ambiguity of the word *bij*, which can 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

(9) Niemand staat boven de wet. [with NP]
Nobody stands above the law.
'nobody is above the law'

(10) Daar gaat niets boven. [stranded]
There goes nothing above.
'that's the best there is'

(11) Ze zijn boven. [intransitive]
They are above.
'they are upstairs'

(12) Olie drijft altijd boven. [particle]
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, but there are also cases in which the gender variation does not correspond to any clear semantic distinction, as in *de/het filter* (the filter) or *de/het soort* (the kind). As a consequence, if such nouns are used without determiner or with a determiner which does not make the gender distinction such as *een* (a/an), it becomes impossible to decide on a specific gender value. In such cases, 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 role of the lexicon

For an automatic tagger it is not strictly necessary to have a pre-defined lexicon, since it can derive one from the training data, but if there is one, it certainly helps, on condition of course that the information which the lexicon provides corresponds to the information that the tagging requires. Once again, such correspondence can usually be taken for granted in systems with a low level of granularity, but not in the fine-grained ones. On average, the more distinctions the tagset makes, the less likely it is to find a lexicon which provides all of the relevant information. At the same time, it would of course be unwise to dismiss the existing lexical resources, such as CELEX and RBN, for the simple reason that they lack information on two or three features. For this reason, we haven chosen for a two-track approach: for the nouns, adjectives, verbs and numerals, we re-use the existing resources, adding some information where needed, and for the other parts of speech (pronouns, articles, prepositions, conjunctions, adverbs), we have designed a new lexicon which provides precisely those distinctions which are needed for the tagging, see (Van Eynde, 1999). At the time of writing, this lexicon has not yet been incorporated in the automatic tagger, but it is used extensively during the manual correction of the tagger output and experiments with the incorporation of lexical resources are going on.

## 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, developed by Polderland BV, and the rule/lexicon-based D-Tale system developed by the Lexicology Group at the Vrije Universiteit Amsterdam. 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 either D-Tale or MBMA.

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

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

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 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, developed by Theo Vosse at Leiden University. 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 (Brill, 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 their 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 Proper-noun, 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. We see that the taggers with fixed tagsets are generally less accurate than the WOTAN based taggers, even though the latter have much larger tagsets, and that among them TnT seems to be 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 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.

These experiments suggest that TnT is the best choice among the available tagger generators, even when only very small amounts of training data are available. However, the accuracy level achieved in this way is still rather low. In experiments that are described in a separate paper (Zavrel and Daelemans, 2000), we have made use of the combination of all available taggers using a second level learner to achieve further error reduction. Training on the larger data sets BATCH-1 and BATCH-2, which seem to be much 'cleaner' data, and contain 22786 and 39304 tokens respectively, results in a final accuracy of 94.3% for the combination. This is a reasonable level of performance for bootstrapping the tagging of the corpus.

## 6. 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. How these problems are being dealt with in the framework of the CGN project has been described in this paper.

## Acknowledgements

This publication was supported by the project "Spoken Dutch Corpus (CGN)" which is funded by the Netherlands Organization for Scientific Research (NWO) and the Flemish Government. For their comments and suggestions we would like to thank Hans van Halteren, Richard Piepenbrock, Lisanne Teunissen, Ineke Schuurman, Ard Sprenger, Antal van den Bosch and the audiences at the first CGN users workshop (Tilburg, August 1999), at the first international CGN workshop (Tilburg, November 1999), and at the CLIN-X conference (Utrecht, December 1999).

| 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, measured on SMALL-1. (MX stands for the MXPOST tagger.)

| | word | D-Tale | KEPER | CORRie | WOTAN-1 MBT | WOTAN-1 MX | WOTAN-1 TnT | WOTAN-2 MBT | WOTAN-2 MX | WOTAN-2 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.

## 7. References


Brants, T., 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*.

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

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

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

Haeseryn, W., 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.

Oostdijk, N., 2000. The spoken dutch coprus project. overview and first evaluation. In *Proceedings of LREC-2000*.

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

Ratnaparkhi, A., 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*.

Van den Bosch, A. 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*.

Van Eynde, F., 1999. CGN lexicon van de functiewoorden. Technical report, CGN-Corpusannotatie. Working Paper.

Van Eynde, F., 2000. Pos tagging en lemmatisering. Technical report, CGN-Corpusannotatie. Working Paper.

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

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

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

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

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