

# DutchSemCor: Building a semantically annotated corpus for Dutch

Piek Vossen<sup>1</sup>, Attila Görög<sup>1</sup>, Fons Laan<sup>2</sup>, Maarten van Gompel<sup>4</sup>, Rubén Izquierdo<sup>3</sup>, Antal van den Bosch<sup>4</sup>

<sup>1</sup>VU University Amsterdam, De Boelelaan 1105, 1081HV Amsterdam, The Netherlands

<sup>2</sup>ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

<sup>3</sup>Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands

<sup>4</sup>Radboud University Nijmegen, P.O. Box 9013, 6500 HD Nijmegen, The Netherlands

E-mail: p.vossen@let.vu.nl, a.gorog@let.vu.nl, fons.laan@gmail.com, proyon@anaproy.nl, r.izquierdo@uvt.nl, a.vandenbosch@let.ru.nl

## Abstract

State of the art Word Sense Disambiguation (WSD) systems require large sense-tagged corpora along with lexical databases to reach satisfactory results. The number of English language resources for developed WSD increased in the past years, while most other languages are still under-resourced. The situation is no different for Dutch. In order to overcome this data bottleneck, the DutchSemCor project will deliver a Dutch corpus that is sense-tagged with senses from the Cornetto lexical database. Part of this corpus (circa 300K examples) is manually tagged. The remainder is automatically tagged using different WSD systems and validated by human annotators. The project uses existing corpora compiled in other projects; these are extended with Internet examples for word senses that are less frequent and do not (sufficiently) appear in the corpora. We report on the status of the project and the evaluations of the WSD systems with the current training data.

**Keywords:** Semantic annotation; Word Sense Disambiguation; Machine Learning

## 1. Introduction

State of the art Word Sense Disambiguation (WSD) systems require large sense-tagged corpora along with lexical databases to reach satisfactory results. While the number of English language resources annotated at the level of lexical semantics has increased in the last decade, data is still scarce for most other languages, Dutch included. In order to overcome the data bottleneck, DutchSemCor<sup>1</sup> is aiming to deliver a one-million word Dutch corpus that is sense-tagged with senses and domain tags from the Cornetto lexical database (Vossen 2006 and Vossen et al. 2007, 2008). The Cornetto database has over 92K lemmas and almost 120K word-senses. It includes both a wordnet and a database with lexical units which provide rich morphosyntactic, semantic and combinatoric information. Synsets in the wordnet part consist of sets of lexical units. The Dutch wordnet is linked to the Princeton WordNet (Fellbaum 1998), SUMO (Niles and Pease 2002) and Wordnet Domains (Magnini and Cavaglià 2000).

In DutchSemCor about 300K examples have so far been manually tagged by two annotators, resulting in 25 examples on average per sense. The examples mainly come from existing corpora collected in the projects CGN (Eerten, 2007), D-Coi, and SoNaR (Oostdijk et al., 2008). These corpora have already been annotated morpho-syntactically in previous projects. In some cases the annotators of DutchSemCor could not find sufficient examples in these corpora for certain word senses. A

web search tool was therefore developed to find additional examples on the Dutch Internet and add these to the data. At the moment of writing, our project is entering the final phase in which the remainder of the corpus will be automatically tagged using different WSD systems. The output of the systems will be validated by human annotators through co-training. When sufficient precision is reached by the WSD systems we automatically annotate the complete corpus not yet manually annotated. In this paper, we describe the project and our approach, and we report on the results so far: both in terms of the manual annotation and the performance of the WSD systems. In Section 2, we describe the work on preparing the corpora. Section 3 describes the manual annotation protocol. In Section 4, we describe the annotation tool that was developed. Finally, in Section 5, we present two WSD systems and their estimated performance.

## 2. Corpus Selection and preparation

The most comprehensive corpus currently available for the Dutch language is the SoNaR corpus. SoNaR is the successor of the D-Coi Project (funded by STEVIN) and aims to contain at least 500 million words of written Dutch. This corpus was selected as the logical primary basis for DutchSemCor. The corpus is fully tokenised, part-of-speech tagged, and lemmatised. Another corpus is CGN which contains about nine million words of transcribed spontaneous Dutch adult speech.

Though SoNaR is large, it still does not contain sufficient examples for certain senses, even though the lexicographers agree it is a valid sense. For this reason,

<sup>1</sup> <http://www2.let.vu.nl/oz/cltl/dutchsemcor/>

the DutchSemCor corpus is augmented with manually selected web snippets. A special web-based tool was developed to allow for the searching of such fragments. Human annotators enter a search query and the system passes the request to a search engine (either mediated through WebCorp.co.uk<sup>2</sup>, or directly). The results are presented on a screen and human annotators select the samples they want to annotate. After selection, snippets are automatically tokenised, part-of-speech tagged and lemmatised using Frog<sup>3</sup> and made available in the corpus annotation tool for assigning the sense.

The final DutchSemCor corpus will thus be a superset of SoNaR, CGN, and the manually-selected Web snippets. We integrated into the corpus representation format FoLiA (Format for Linguistic Annotation<sup>4</sup>) the ability to annotate lexical semantic senses, along with their annotators and confidence.

### 3. Manual annotation

The DutchSemCor corpus is split into two parts that are handled in different ways. The first part of about 300,000 word tokens is annotated manually in a traditional way (compare OntoNotes & SemCor): a group of 8 human annotators analyzed and tagged an average of 25 examples per sense of the 3,000 most frequent and most polysemous words of the Dutch language (65% nouns, 23% verbs and 12% adjectives). The procedure was supported by a knowledge-rich tagging system (SAT, see next section).

During manual annotation, two annotators consider the same lemmas and KWIC index examples of the reference corpus to annotate. Each tagged sentence and every annotator action is recorded in a separate database. From the database we regularly derive the annotation statistics and status (see Figure 1). The table shown in Figure 1 contains information and scores for each annotated word, such as number of annotators, number of senses, number of annotations, overlap, agreement, and proportion of annotation per sense. The total agreement/disagreement proportion per word results in the overall Inter-annotator Agreement (IA) which is our quality measure. If the IA is less than 80%, annotators examine the disagreements and improve the annotations until an IA of 80% or higher is reached.

In previous projects such as OntoNotes (Sameer and Nianwen, 2009) similar cycles have been used to reach high IA scores. To our knowledge no further criteria have been applied in these projects. Our aim is to not only obtain an IA score of 80% or higher, but also to deliver a large corpus that is sufficiently diverse in terms of syntactic and semantic patterns. We are trying to reach high diversity by implementing different filters which make use of constituency patterns, semantic roles, collocational information, and domain labels. This way we not only guarantee rich and interesting data for purposes of linguistic research but also a semantic corpus with optimal variation for machine learning. Text fragments with a large syntactic and semantic diversity can better serve WSD techniques and yield better results when used for bootstrapping.

In order to ensure an optimal coherence in the annotation we have frequent meetings with the annotation team. In these meetings we reflect on problems of different origins (possible mistakes in the lexical database, difficult sense distinctions, senses not represented in the corpus). We also discuss co-occurrence strategies to find word meanings directly in the corpus or on the Internet as well as to group examples and to discover figurative and idiomatic uses. Another purpose of the discussions is to gain insight into the peculiarities of the Dutch language and to teach annotators to validate their language instincts using different word meaning tests (e.g. zeugma, cross readings). In the initial phase, these meetings were held bi-weekly for reasons of training and tool-testing. At present, they take place once a month.

Current results of manual annotation:

- PoS: nouns, verbs and adjectives
- number of annotated lemmas: 2,589
- number of word senses: 10,172
- number of overlapping annotations<sup>5</sup>: 255,625
- IA<sup>6</sup>: 93%
- Coverage 1<sup>7</sup>: 77%
- Coverage 2<sup>8</sup>: 86%.

<sup>2</sup> <http://www.webcorp.org.uk/>

<sup>3</sup> <http://ilk.uvt.nl/frog>

<sup>4</sup> <http://ilk.uvt.nl/fofia> FoLiA is based on the D-Coi XML format, but introduces a universal paradigm allowing for various kinds of linguistic annotation; including lexical semantic sense annotation. FoLiA is also proposed as a CLARIN-NL standard in the context of the TTNWW project, and adopted in other projects as well.

<sup>5</sup> Tokens annotated by two annotators

<sup>6</sup> Inter-Annotator Agreement (also referred to as IAA)

<sup>7</sup> Proportion of senses with 25 or more annotations

<sup>8</sup> Proportion of annotations given 25 tokens per sense required

3	OVERVIEW									
4	POS	verb								
5	Nr of words	681								
6	Nr of completed words	278	0.40822320117474303%							
7	Proportion of senses with	46.366.288.899.009.500	0.6808559309693025%							
8	Proportion of annotations	50.790.877.003.062.300	0.7458278561389472%							
9										
10	Word	Annotators	Nr of senses	Nr of annot	Nr of annot	Overlap	Average IAA	Sense proporti	Annotation prop	Sense distribution
11	afschieten	Jonica;Lisanne;	7	2	86	82	100	0.428571428571	0.44	cover:25:27:27:2:0:
12	openen	Jonica;Lisanne;	4	2	122	118	97	1.0	1.0	cover:28:28:32:26:
13	invoeren	Elizabeth;Marlisa;	4	2	113	113	96	1.0	1.0	cover:29:25:28:27:
14	scheiden	Daphne;Wilma;	5	2	156	148	99	0.8	0.8	cover:36:36:46:29:
15	uitslaan	Jonica;Lisanne;	7	2	224	219	98	1.0	1.0	cover:31:37:27:30:3:
16	inspelen	Elizabeth;Marlisa;	4	2	131	119	100	1.0	1.0	cover:26:36:29:30:
17	voldoen	Elizabeth;Marlisa;	3	2	104	104	100	1.0	1.0	cover:39:29:37:
18	verrijden	Anneleen;Charlotte;	3	2	95	95	100	1.0	1.0	cover:39:30:26:
19	afschieden	Jonica;Lisanne;	3	2	96	96	100	1.0	1.0	cover:30:30:36:
20	knakken	Elizabeth;Marlisa;	4	2	101	101	100	0.5	0.88	cover:26:38:17:21:
21	scheppen	Elizabeth;Marlisa;	6	2	166	165	98	0.833333333333	0.92	cover:30:30:29:34:2:
22	doorsteken	Jonica;Lisanne;	5	2	140	128	100	0.8	0.992	cover:27:24:26:25:2:
23	doortrekken	Jonica;Lisanne;	7	2	199	196	100	0.857142857142	0.92	cover:33:26:38:27:2:
24	afronden	Jonica;Lisanne;	3	2	48	47	100	0.0	0.626666666666	cover:24:9:14:
25	koken	Jonica;Lisanne;	4	2	100	97	96	0.75	0.83	cover:26:26:33:8:
26	afwerken	Jonica;Lisanne;	3	2	84	81	100	1.0	1.0	cover:29:27:25:
27	verkopen	Jonica;Lisanne;	4	2	103	102	100	0.5	0.64	cover:61:1:27:13:
28	richten	Jonica;Lisanne;	6	2	149	149	99	0.833333333333	0.853333333333	cover:33:3:29:29:27:
29	neerkomen	Elizabeth;Marlisa;	3	2	111	107	100	1.0	1.0	cover:47:27:33:
30	nemen	Elizabeth;Marlisa;Daphne	8	3	222	129	64	0.0	0.565	cover:18:19:18:18:1:
31	vergoeien	Jonica;Lisanne;	4	2	72	72	100	0.5	0.56	cover:27:39:6:
32	versieren	Daphne;Wilma;	3	2	134	134	99	1.0	1.0	cover:48:53:33:
33	aanspreken	Jonica;Lisanne;	4	2	122	117	100	1.0	1.0	cover:33:31:27:26:

Figure 1: Logfile converted into feature table

#### 4. Semantic Annotation Tool (SAT)

The SAT<sup>9</sup> is a web application for semantic tagging developed for DutchSemCor. The SAT user interface (see Figure 2) combines lexicographic information from the Cornetto database (in the top table) with corpus data from SoNaR (in the bottom table). For each lemma lexicographic and corpus data are retrieved. For each sense of the lemma the annotator selects the corpus lines that apply (the blue lines in the top and bottom tables in the screenshot). The combinations of word sense and applicable corpus lines are saved in a database, and the process is repeated until a sufficient number of instances in the corpus are annotated for each sense.





To ease the finding of required contexts, the SAT allows co-occurrence filtering of arbitrary words in the left and right context of the lemmas (Figure 3).

<sup>9</sup> The *Manual for Semantic Annotation* contains a more detailed description from the user perspective, together with accompanying screenshots. The SAT tool can be viewed at: <http://cornetto.science.uva.nl:8080/dutchsemcor/>

Mode: Free List Buddy Lemma:  Category: adj noun verb ? Context:  chars Source: All SoNaR Snippets CGN DB Search

#	Examples	Morphosyntax	Resume/Def	Domain	SUMOntology	Synonyms	Relati
1	beton zet uit	v-intr-sch-nrefl	zwellen	alg	Increasing Orga	opzwellen uitdijen zwellen	verar
2	de radio <b>uitzetten</b>	v-tr-sch-nrefl	afzetten	ind	Motion	afzetten uitschakelen	uitdoe
3	vreemdelingen <b>uitzetten</b>	v-tr-sch-nrefl	uit het land zetten	biol med alg	Removing Expre	uitwijzen	wegb
4	een speurtocht <b>uitzetten</b>	v-tr-sch-nrefl	afbakenen	alg	IntentionalProce		afbak
5	vis <b>uitzetten</b> in een vijver	v-tr-sch-nrefl	ergens doen verspreiden	alg	Putting		versp
6		v-tr-sch-nrefl	op interest plaatsen	ec	Investing	plaatsen	onde

1 of 6 rows

 Tag Co-oc L:  M:  R:   Clear  Filter  UnTag Usage: N F I U 1 ... 1

#	tfel	Left	Sense	Sense ids	Word	Domain	Right	date time
166		neknarf itsluitend in dollars , ponden of Zwitserse franken			uitzetten		, namelijk stuk voor stuk gemiddeld onder nul . Fe	2004.01.10 00:00
10		tad garde van n jaar en bij rentevoet p is het bedrag dat	6	r_v-8638	uitgezet	ec	tegen samengestelde interest bij de genoemde re	2009.01.27 00:00
26		nijz nelluten dat er over 10 jaar , 400 antilopen zullen zijn			uitgezet	biol	. Oog in oog met je idool . Dat konden fans op de	2009.01.27 00:00
172		gnimrawven oude houtwerk in de kerk zou bij verwarming			uitzetten		en breken . Dus diepe koude moet . Zetten popar	2003.04.26 00:00
50		etuur een . Met rode linten heeft de organisatie de route			uitgezet		waar de tunnel moet komen . Het gebied was het	2001.11.08 00:00
163		hcot netplantages de doorslag en werden de dieren toch			uitgezet		. Voorbeelden zijn : X Kritiek van zowel bijenhou	2009.01.27 00:00
162		nereid e Dat vind ik wel heel knap . Nu gaan we de dieren			uitzetten		aan de overkant van de weg . De diertjes word	2009.04.30 00:00
164		nereid e . Inmiddels zijn in gevangenschap gefokte dieren			uitgezet	biol	in Wyoming . De zwartvoetbunzing leeft voornar	2009.01.27 00:00
101		gitamdrijf met ratelen is gestopt , de machine handmatig			uitzetten		. Het gaat erom dat je de afsluitprocedure doorlo	2009.01.27 00:00
31		nedrow inetisch veranderde muggen in de natuur worden			uitgezet	biol	. Maar dat is niet de enige bestrijdingsmethode di	2009.01.27 00:00
28		tfesh dearden die Natuurmonumenten in het gebied heeft			uitgezet		. De vraatzucht van de beesten zorgt voor het al	2004.03.18 00:00
11		tdrow tekening wordt afgegeven , maar op krediet wordt			uitgezet	ec	. Ik denk dat dat nu juist de aanleiding vormt voo	2009.01.27 00:00
103		raamoz 'aaron mag je een Windows-machine niet zomaar			uitzetten		zonder de afsluitprocedure doorlopen te hebben	2009.01.27 00:00
142		koo rethn instellen . Deze statuscellen kunnen echter ook			uitgezet		worden , waardoor alle braillecellen voor het lees	2009.01.27 00:00

1 of 178 rows

Figure 2: SAT interface

	Morphosyntax	Resume/Def	Domain	Synonyms
ver [de verkiezingen]	n-het-t	te publiceren stuk	media	krantenartikel
e <b>artikelen</b>	n-het-t	te verhandelen voorwerp	handel	handelsartikel
d 1 van het Burgerlijk Wetboek	n-het-t	onderdeel v.e. wettekst	jur	wetsartikel
evoeegen aan het woordenboek	n-t	eerste woord van een artikel in e taal		lemma
n met een <b>artikel</b> zijn naamwoordsgroepen.	n-t	woordsoort die uitsluitend met e taal		lidwoord

Co L:  M:  R:  Clear Filter UnTag Usage: N F I U

	Sense	Word	Right
ge salaris wat ze krijgen . en de Kasteelreeks en zo . ik heb een heel <b>leuk</b>		<b>artikel</b>	over dat soort boeken <b>gelezen</b> in 't Russisch .
Sportzomer <b>Leuk</b>		<b>artikel</b>	over de ` spetterende sportzomer ' ( AD , 16

**SoNaR context of row 1 (1)**

voor dat lage salaris wat ze krijgen . en de Kasteelreeks en zo . ik heb een heel **leuk** **artikel** over dat soort boeken **gelezen** in 't Russisch . omdat die boeken bestonden vroeger domweg in 't Russisch niet mocht niet . en

Figure 3: SAT co-occurrence filtering

## 5. WSD systems

Word sense disambiguation (WSD) is one of the target application areas of the DutchSemCor corpus, but it is also used for its creation. In the second phase of the project, we apply WSD methods to the corpus using the annotations that have been carried out in the first part. In fact, we apply a number of different methods:

- Knowledge-based WSD that employs the relations from the Cornetto database and in some cases from the English WordNet.
- Supervised machine learning-based WSD that creates word experts from annotated examples
- Named Entity recognition and Wikification

Named Entity recognition and Wikification are carried out independently of the Cornetto database and applied to the complete corpus. Each Named Entity will receive a link to the corresponding Wikipedia page if present. Besides representing a separate semantic annotation, the Named Entities can also be used as features for WSD.

In this paper, we focus on the first two approaches. For the knowledge-based WSD we use the UKB system that was developed by Agirre and Soroa (2009). UKB considers wordnet as a graph, where synsets are the nodes and the relations between synsets are edges. It applies a page-rank algorithm to calculate the weight for each synset (a node) in the graph. To disambiguate a new word, the personalized page-rank algorithm implemented in UKB activates the synset nodes of words that occur in the context of the focus word, and then propagates the weights from these activated synsets, resulting in a score for all the target word senses.

The supervised WSD system uses memory-based machine learning techniques implemented in TiMBL (Daelemans *et al.*, 2007) to build word experts, each responsible for disambiguating the senses of one of the designated target words in this project. The word experts base their decision on both local context, such as neighbouring words, and more global context, such as predictive words occurring in neighbouring sentences (Hoste *et al.*, 2002; Decadt *et al.*, 2004).

In the next sections, we describe both systems in more detail, and compare their performance.

### 5.1 UKB results

UKB requires a lexicon of lemmas with pointers to concepts and a data file with relations between concepts from which a graph is built. The Dutch lexicon contains about 84,000 lemmas that map to about 70,000 synsets. Table 1 shows the static semantic relations that have been used to build graphs for the UKB. The Dutch synset relations (DS:DS) are EuroWordNet relations (Vossen 1998). The synset-domain relations (DS:DO) originate from WordnetDomains and have been imported through the equivalence relations with the

English WordNet to the Dutch synsets. We also included the domain hierarchy itself (DO:DO relations) as relations. Likewise, synsets for *tennis player* and *tennis ball* are related to the domain *tennis* but since the domain *tennis* is linked to the domain *sport*, the *tennis* synsets are indirectly related to synsets for *football player* and *football*, since the latter are related to the domain *soccer* which is also related to the domain *sport*. In case there is an equivalence relation between the Dutch synsets and the English WordNet, these are also presented as relations in the UKB (DS:ES). Finally, we have the relations from the English WordNet itself, both the direct relations (ES:ES) and the relations from a synset to the disambiguated glosses (ES:EG). In total, almost 1 million static semantic relations are available.

Type of relation	Relations
DS:DS, Dutch_synset/Dutch_synset	140,219
DO:DO, Domain/Domain	125
DS:DO, Dutch_synset/Domain	86,798
DS:ES, Dutch_synset/English_synset	73,935
ES:ES, English_synset/English_synset	252,392
ES:EG, English_synset/English_gloss_synset	419,387
	972,856

Table 1: Semantic relations used for the UKB

Many annotations of words in DutchSemCor occur in the same sentence. By assuming that these synsets are somehow semantically related, we can derive many new relations from the annotations. We extracted two sets: different polysemous words annotated in the same sentence and annotated polysemous words that co-occur with words that have a single meaning. This adds another 168K relations to the graph (see Table 2).

	Sentences	Relations	Overlap
<b>Polysemous words</b>	18,653	17,152	2,644
<b>Monosemous words</b>	189,411	151,598	3,471

Table 2: Semantic relations derived from the annotations

The co-occurrence relations hardly overlap with the relations already present in the Dutch wordnet: 2,644 polysemous word relations (15%) and 3,471 monosemous word relations (2%) were already present in the static relation set.

For determining the relevance of a relation, we cannot use the direct frequency since it is bound by the number

of annotations per sense (25 on average).<sup>10</sup> Most relations occur only once. To still assign a weight to the extracted relations we calculated the average information value for each relation, where the information value  $I$  for a synset  $s$  is determined by the number of relations in which it occurs in the extracted set divided by the different synsets to which it is related:

$$I(s) = \frac{N(s)}{N(t)}$$

$N(s)$  stands for the number of relations in which a synset  $s$  occurs and  $N(t)$  stands for the number of target synsets it is related to. We derive the average information value  $AvgI$  for a relation  $r$  as the sum of the information value of the two related synsets, divided by 2. The  $AvgI$  is added to the relations imported into the UKB graph. Inspection of the highest scoring relations showed many good conceptual relations. For example, we find among the polysemous words relations between *konig* (king), *konigin* (queen), *paard* (horse), *loper* (bishop), *toren* (tower), *stuk* (chess piece) and *slaan* (take a chess piece) all in their chess meaning.

We built 5 different graphs using the following relations:

- UKB1: DS:DS+DO:DO+DS:DO
- UKB2: UKB1+DS:ES
- UKB3: UKB2+ES:ES+ES:EG
- UKB4: UKB1+poly+mono
- UKB5: UKB3+poly+mono

## 5.2 Knowledge-based WSD results

Table 3 shows the results of evaluating the different graphs on a test set of 35,269 tokens (both nouns and verbs) extracted from the annotated data (see below for more details on the test set). The UKB has different methods for exploiting the graph. Our experiments so far showed that the personalized page rank considering each word separately (ppr\_w2w setting) gave the best results.

UKB5 which uses all the relations has the best scores for both precision and recall. UKB4 comes very close, however, without using English (equivalence) relations. Actually, we see that adding the new relations derived from the annotations boosted the results with almost 9%. This suggests that the number of relations is, in fact, more important than the careful manual selection of relations. The fact that we find more syntagmatic relations in the annotations than paradigmatic relations from the wordnets is also very likely to play a role. Thus, when more data is annotated we can also increase the relations to be added and derive more statistical information on the strength of a relation (which is now limited by the maximum of 25 examples per sense).

<sup>10</sup> Note that we will be able to extract these statistics when the complete corpus is tagged with sufficient precision by the WSD system.

	Precision	Recall	F-measure
UKB1	0.4557	0.4491	0.4523
UKB2	0.4557	0.4491	0.4524
UKB3	0.4560	0.4493	0.4526
UKB4	0.6360	0.6272	0.6316
UKB5	0.6411	0.6322	0.6366

Table 3: Evaluation results of the Knowledge-based WSD by UKB

Earlier versions of the Dutch UKB1 and UKB3 were evaluated in the SemEval2010 task on Domain Specific WSD (Agirre *et al.*, 2010). UKB3 performed best with a precision of 52,6%. For comparison, the English UKB scored a precision of 48,1% on the English task and ranked 10<sup>th</sup> among all participating systems. UKB3 performs 7% lower in our evaluation due to the fact that our test is more difficult: it is a sample-based evaluation for the most polysemous words only, whereas the SemEval2010 task was an all words task for a specific domain. In the latter case, there are more domain specific monosemous and low-polysemous tokens in comparison to our test.

## 5.3 Memory-based word experts

The supervised machine learning-based WSD system employs  $k$ -Nearest Neighbour classifiers (Aha *et al.*, 1991) for word sense disambiguation. Our current approach follows previous research by Decadt *et al.* (2004) and Hoste *et al.* (2002). Each classifier constitutes one word expert, and each word expert disambiguates between the senses of one of the target words selected for the project. We first illustrate the working of the system. Given the corpus and the annotated data gathered by the annotators, two datasets can be extracted: a training set and a test set. Recall that the project aims to manually annotate 25 examples per sense. Of these 25 examples, 10 are selected to be included in the test set while the remaining 15 (or more if more than 25 examples were annotated) are included in the training set. All examples that are included in either set have an IA above the predefined threshold of 80%, and the minimum number of examples per senses is satisfied for each sense of the word under consideration. Words that do not fit these requirements are ignored. These split sets are only used for evaluation purposes. When the system is run on the remainder of the corpus to automatically annotate previously unseen examples, the full 25+ examples per sense are used for training the system.

To run the memory-based WSD system, training and test instances are extracted from the corpus for each word expert, each instance being one occurrence of the target word, sense annotated by a human annotator. These instances consist of a feature vector and a class label, the latter being the sense (i.e. lexical unit ID as defined in Cornetto). If no test set is used for evaluation, test instances are simply all previously unseen instances in the corpus, without associated class label, as there is no sense known prior to classification.

The feature vector consists of three components: a local context part including the word itself, a global context part and, optionally, a domain label if present. The local context part, in turn, consists of a certain number of words to the left of the word under consideration followed by the word itself and a certain number of words to the right. The context sizes, left and right, are adjustable parameters. In addition, the local context part of the feature vector may be enhanced with linguistic features; the corpus contains data on part-of-speech tags and lemmas that may be included in the feature vector for each word in context. These too are parameters to the system for which the optimal settings can only be found experimentally.

The global context part of the feature vector consists of binary bag-of-word features in which the presence or absence of important predictor words in the same sentence of the sample word is flagged. The global part refers to the fact that a certain word can be an important predictor for a given target lemma and sense and that it is computed globally over the corpus as a whole according to the method put forward by Ng et Lee (1996).

The machine learning algorithm used is implemented in the TiMBL software package (Daelemans et al., 2007) which is called by the supervised WSD system to train and test the word-expert classifiers. TiMBL is governed by several hyper parameters to tune the classifier performance. A key parameter for  $k$ -Nearest Neighbour classification is the value of  $k$ . Finding optimal parameters for a particular classifier is an experimental process in which ideally all interdependent parameter combinations are tested. In the supervised WSD system, we perform automated parameter optimisation for TiMBL on a per-classifier basis. Thus, for each word expert, prior to testing, optimal parameters are sought using a pseudo-exhaustive test of different hyper parameter setting combinations tested using leave-one-out cross-validation on the training data.

#### 5.4 Supervised WSD-results

For the evaluation of the WSD system, we selected all words from the annotated part of the SoNaR corpus that had at least 25 agreed annotated instances per sense. We trained the word-expert for that word with all annotated

instances, splitting the instances for each sense into 10 testing and at least 15 training examples.

Table 4 shows the performance in terms of token accuracy of the supervised WSD trained with different feature sets and evaluated over the test set. The training and test sets were generated from the annotated part of SoNaR at an early stage of the project, containing only 11,292 tokens. The size of the context window is shown for each type of feature as subscript. Two baselines are also included, one following a random heuristic and the other selecting the first sense based on Cornetto.

Feature set	Token accuracy
<i>Chance Baseline</i>	<i>0.2736</i>
<i>First sense baseline</i>	<i>0.2765</i>
Words <sub>1</sub>	0.6287
Words <sub>1</sub> + Lemmas <sub>1</sub>	0.6343
Words <sub>1</sub> + PoS <sub>1</sub>	0.6307
Words <sub>1</sub> + Lemmas <sub>1</sub> + PoS <sub>1</sub>	0.6333
Words <sub>2</sub>	0.6511
Words <sub>2</sub> + Lemmas <sub>2</sub>	0.6486
Words <sub>2</sub> + PoS <sub>2</sub>	0.6393
Words <sub>2</sub> + Lemmas <sub>2</sub> + PoS <sub>2</sub>	0.6409
Words <sub>3</sub>	0.6606
Words <sub>3</sub> + Lemmas <sub>3</sub>	0.6535
Words <sub>3</sub> + Bag-of-word	0.7212
Words <sub>4</sub>	0.6551
Lemmas <sub>4</sub>	0.6574
Words <sub>4</sub> + Lemmas <sub>4</sub>	0.6475
Words <sub>5</sub>	0.6503
Words <sub>6</sub>	0.6467
Words <sub>7</sub>	0.6438
Words <sub>8</sub>	0.6425
Words <sub>9</sub>	0.6395

Table 4: Performance of the supervised WSD

In general, the effect of considering a wider context does not have significant impact on the performance of the system. The same situation applies when enriching the set of features with part-of-speech tags and lemmas. The behaviour is not always as we would expect and the performance is not higher in all cases when a richer set of features is selected.

We also generated another training and test set in a more advanced stage of the annotation process. The number of tokens was 35,338. The polysemy distribution of the test set can be seen in figure 4. As said before, we do not

consider monosemous words in our corpus. Most words in the test set have two or three senses.

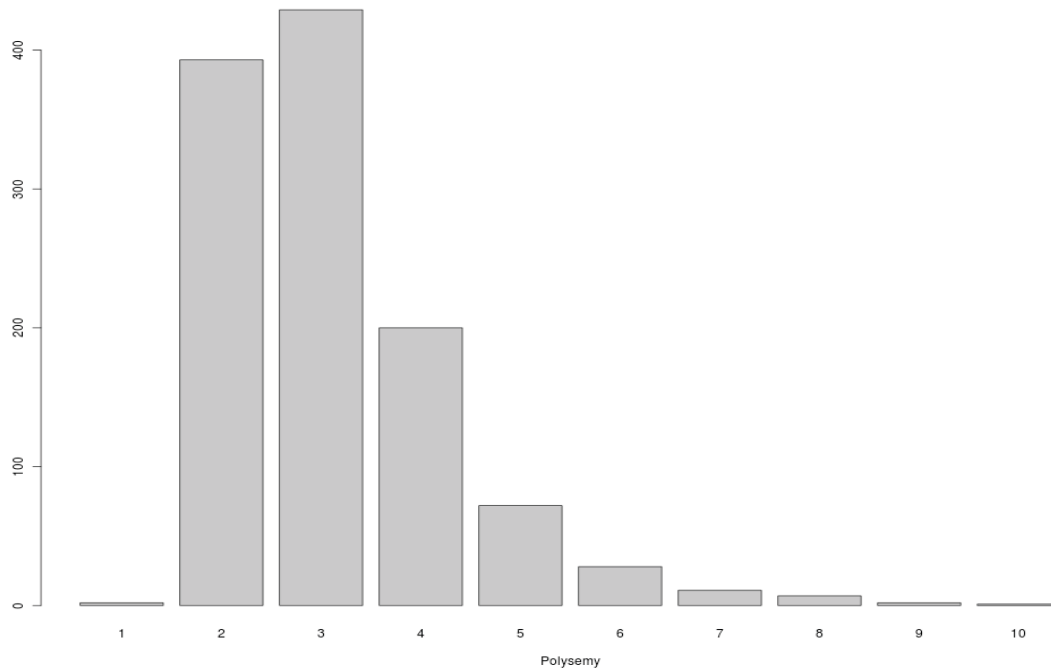


Figure 4: Polysemy of the test set

Table 5 again shows the performance of the system, using the newer data sets. In this experiment we assess the impact of the bag-of-words features and parameter optimisation.

Feature set	Token accuracy
Words <sub>1</sub>	0.6462
Words <sub>1</sub> + Bag-of-words	0.7259
Words <sub>1</sub> + PoS <sub>1</sub> + Bag-of-words	0.7226
Words <sub>1</sub> + Bag-of-words + PS	0.7931

Table 5: Performance on newer data sets

As we can see, the use of bag-of-words sets leads to an important improvement of around 8% in token accuracy. On the contrary, the part-of-speech tag seems not to help the classification at all, not providing any advance. Last, we can see that using the parameter optimisation search

(PS) for TiMBL, results can be improved with another 7%. The best performance 0.79 scores considerably higher than the knowledge-based UKB5 (0.64). This is also known from all earlier WSD evaluations in Senseval and Semeval. For future evaluations, we will create an all-words test set independent of the selected corpus to better test the systems independently of the corpus. We will also see how the two systems can complement each other.

Figure 5 shows the evolution of our system regarding the confidence assigned by the TiMBL engine to each token. In the standard evaluation, we considered for each test instance, the sense proposed by TiMBL, regardless the confidence assigned to it. We made an analysis of how good the confidence value was by filtering out instances with a confidence under a threshold. We expected the discarded instances to remain untagged, the recall to be lower and the precision to be higher.



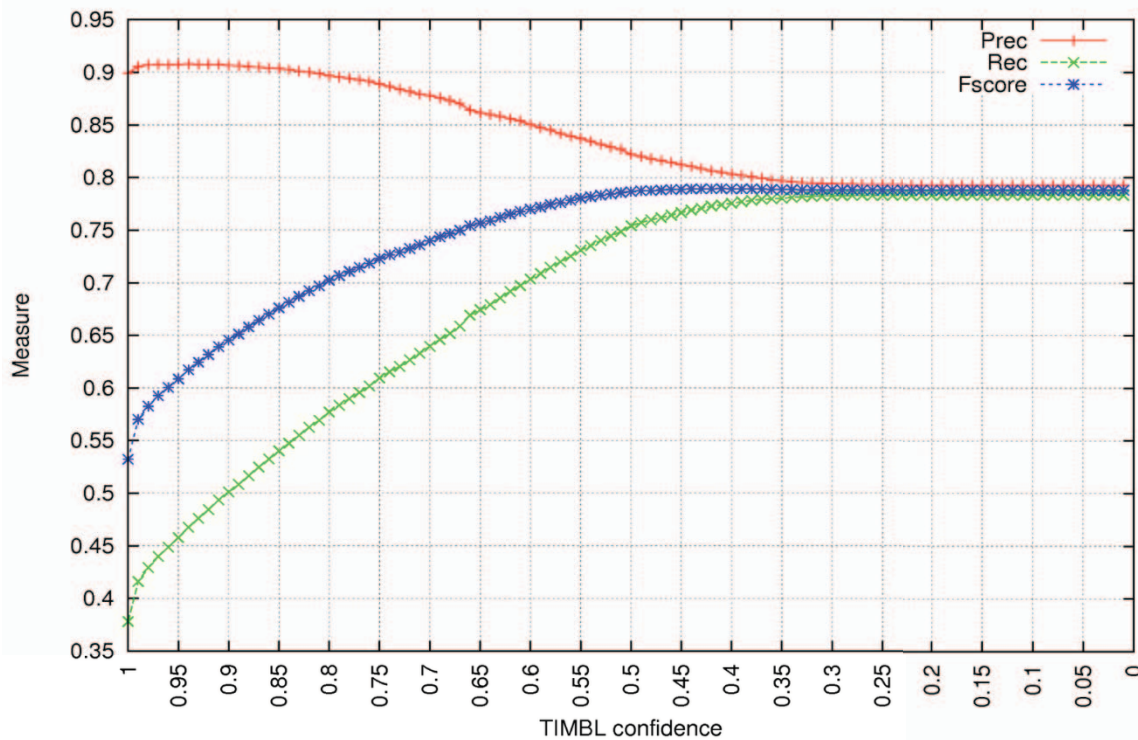


Figure 5: TiMBL confidence

The results were as expected. When we used a high TIMBL confidence value, the precision was indeed very high while the recall was hardly penalized. As we chose lower values for the confidence threshold, both values tended to be similar. It is worth mentioning that selecting a confidence of 0.55 for TIMBL results in the precision of 0.8370 (+0.439 compared with no filtering) and in an Fscore of 0.7804 (only -0.027 less than no filtering). We can filter the test instances according to the threshold of 0.55, improving the precision with 4 points and without losing too much recall.

In future experiments, it might be interesting to find out how the system performs for individual words in comparison with the its global accuracy. In spite of a good overall performance, it can be the case that the system works with high precision for certain words but reaching low results for other lemmas.

Figure 6 shows the number of words for which the system reaches a certain accuracy. Considering a quite high and reasonable minimum accuracy of 0.8, 65.54% of the nouns obtain a higher accuracy, and only the 31.21% of the verbs exceed this threshold. The manual annotation of verbs is still ongoing so these numbers are expected to increase.

## 5.5 Co-Training

The next phase in the project will consist of co-training. The procedure is as follows:

1. Train the WSD system with the current data (minus the test set) and determine the accuracy for each word and the F-measure for each word meaning.
2. Select which words perform with accuracy below 80% in the evaluation. This is the co-training word set  $W_{co}$ . Words that already perform well are ignored.
3. Apply the WSD systems to all occurrences of  $w_i$  element of  $W_{co}$  that have not been annotated yet.
4. We select the corpus sentences  $S$  in which the WSD assigned a sense  $c$  of  $w_i$ , such that  $c$  has an F-measure below 80% in the evaluation. Sentences with good performing meanings are ignored.
5. We determine a co-training score for each of sentence  $s$  in  $S$ .
6. We load the top-200 sentences into the annotation tool with the meaning assigned by the system as if it was an annotator.
7. The human annotators check the sentences assigned by the system and confirm or correct them.

8. After a week, we add the checked examples to the data to improve the WSD system and return to step 1.

The co-training score for each sentence is based on the confidence of the WSD system and the distance score of the TiMBL system. We select sentences with a high

score and high distance. These are examples that are very different from the examples of the training set but for which the system nevertheless has strong evidence for the meaning. We want the students to find very different sentences for weak meanings but need to be sure that the sentences are relevant to that meaning.

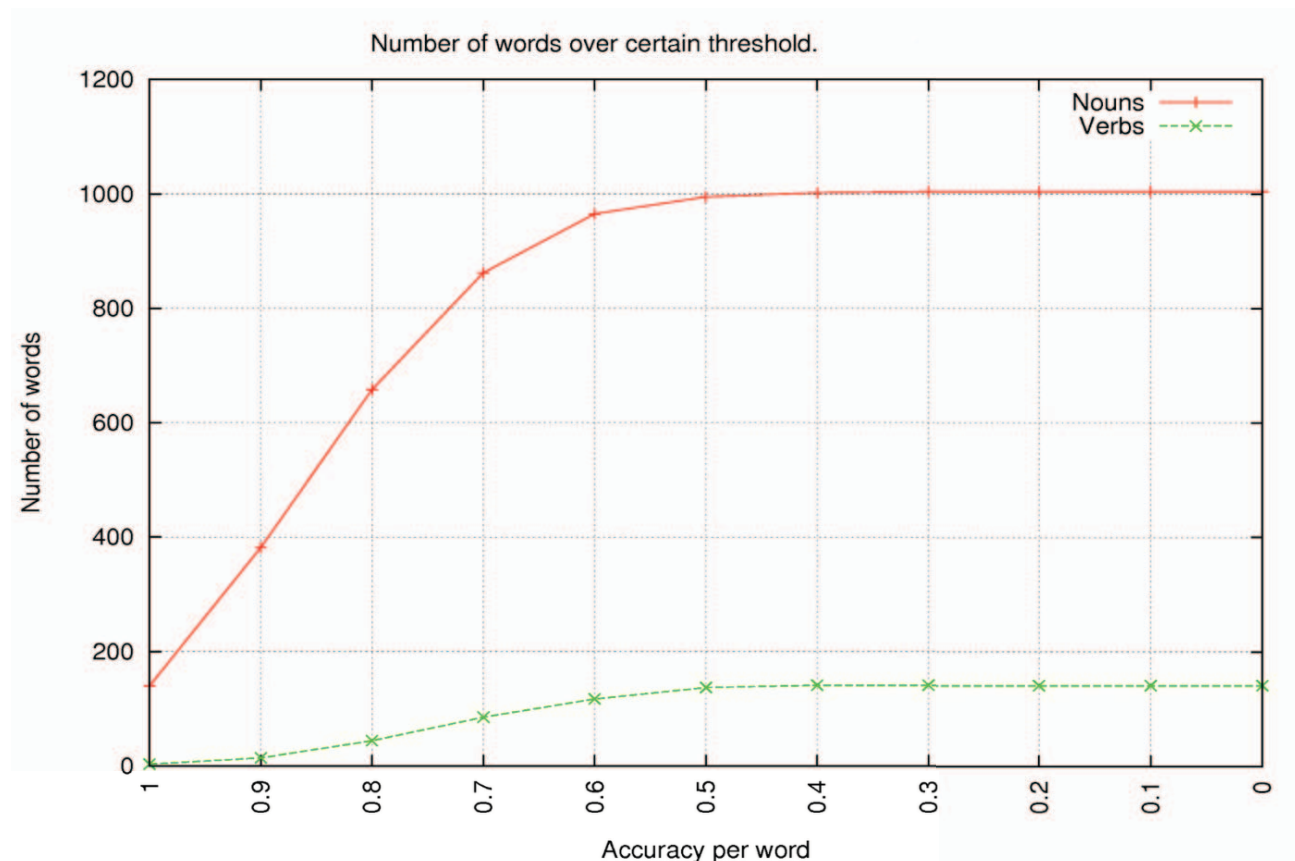


Figure 6: Number of words with a certain accuracy

We will repeat the cycles until we reach 80% accuracy for all the 3,000 words. When sufficient quality of the WSD is reached, we apply WSD to the whole corpus. The TiMBL system can only assign senses to the trained words (3,000). UKB can assign senses to all words in Cornetto. Monosemous words can simply be tagged. We will also experiment with combinations of WSD systems.

## 6. Conclusion

In this paper, we have described the different phases of the DutchSemCor project: from manual annotation to the use of different WSD systems. We discussed the selection and processing of the SoNaR corpus as well as the working methods and tools used throughout the manual annotation phase. We showed that the development of WSD systems is not only a goal of the project itself but is also necessary for providing correct annotations for the complete corpus. We have seen that, even though the initial results of the two WSD systems

are promising, there is still ample space for fine tuning the software through experimentation. Finally, we have summarized our future plans for Co-Training, which will take place in the coming months.

## 7. Acknowledgements

The DutchSemCor project is an NWO<sup>11</sup> Humanities "Middelgroot" (Medium) Investment subsidy project with a subsidy period of September 2009 - August 2012. We would like to thank NWO for making it possible to carry out the project.

## 8. References

- Aha, D.W., Kibler, D. & Albert, M.K. (1991). Instance-Based Learning. AITaking into account these and previous results, we select the feature set of the last experiment (Words1 + Bag-of-words + Parameter Search), as the configuration to build our first version

<sup>11</sup> <http://www.nwo.nl>

- of the WSD system. Further experimentation will be carried out using this system. *Journal of Machine Learning*, 1, pp. 37-66.
- Agirre, E., Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of EACL-09*, pp. 33-41.
- Agirre, E., Stevenson, M. (2006). Knowledge sources for WSD. In *Word Sense Disambiguation: Algorithms and Applications*. New York, NY: Springer, pp. 217-251.
- Agirre E., Lopez de Lacalle, O., Fellbaum, C., Hsieh, S., Tesconi, M., Monachini, M., Vossen, P., & Segers, R. (2010). SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. In K. Erk, C. Strapparava (eds.) *Proceedings of SemEval-2010: 5th International Workshop on Semantic Evaluations on Kyoto's subtask WSD17: All-words Word Sense Disambiguation on a Specific Domain, workshop collocation: ACL2010*, July 11-16, 2010, Uppsala, Sweden: The Association for Computational Linguistics (ACL), pp. 75-80.
- Daelemans, W., Zravel, J., van der Sloot, K. & van den Bosch, A (2007). TiMBL: Tilburg Memory Based Learner, version 6.1. Reference Guide. ILK Technical Report 07-07.
- Decadt, B., Hoste, V., Daelemans, W. & van den Bosch, A. (2004). GAMBL, genetic algorithm optimization of memory-based WSD. In R. Mihalcea, P. Edmonds (eds.) *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, Barcelona, Spain, July 2004, pp. 108-112.
- Eerten, L. (2007). Over het Corpus Gesproken Nederlands. In *Nederlandse Taalkunde*, 12(3) pp. 194-215.
- van Gompel M. UvT-WSD1: A cross-lingual word sense disambiguation system. In *SemEval'10: Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden, pp. 238-241.
- Hoste, V., Hendrickx, I., Daelemans, W. & van Den Bosch, A. (2002). Parameter optimization for machine-learning of word sense disambiguation. *Nat. Lang. Eng.* 8(4), pp. 311-325.
- Kilgariff, A. (2006). Word senses. In *Word Sense Disambiguation: Algorithms and Applications*. New York, NY: Springer, pp. 29-46.
- Magnini, B., Cavaglia, G. (2000). Integrating Subject Field Codes into WordNet. In M. Gavrilidou, G. Crayannis, S. Markantonatu, S. Piperidis & G. Stainhaouer (eds.) *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*. Athens, Greece, 31 May - 2 June, 2000, pp. 1413-1418.
- Mihalcea, R. (2002) Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Language Resources and Evaluations (LREC 2002)*, Las Palmas, Spain, pp. 1407-1411.
- Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of the 8th Conference on Computational Natural Language Learning CoNLL*, Boston, MA, pp. 33-40.
- Navigli, R. (2009). Word Sense Disambiguation: a Survey. In *ACM Computing Surveys*, 41(2), ACM Press, pp. 1- 69.
- Ng, H.T. (1997). Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, USA, pp. 1-7.
- Ng, H.T., Lee, H.B. (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics (ACL '96)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 40-47.
- Oostdijk, N., Reynaert, M., Monachesi, P., van Noord, G., Ordelman, R., Schuurman, I. & Vandeghinste, V. (2008). From D-Coi to SoNaR: A reference corpus for Dutch. In: *Proceedings on the sixth international Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.
- Palmer, M., Ng, H.T. & Dang, H.T. (2006). Evaluation of WSD systems. In *Word Sense Disambiguation: Algorithms and Applications*. New York, NY: Springer, pp. 75-106.
- Pianta, E., Bentivogli, L. (2003). Translation as Annotation. In *Proceedings of the AI\*IA 2003 Workshop 'Topics and Perspectives of Natural Language Processing in Italy'*. Pisa, Italy, pp. 40-48.
- Sameer, S.P., Nianwen, X. (2009). OntoNotes: the 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*. Boulder, Colorado: Association for Computational Linguistics, pp. 57-60.
- Vossen, P. (2006). Cornetto: Een lexicaal-semantische database voor taaltechnologie, *Dixit Special Issue*, Stevin.
- Vossen, P., Hoffman, K., de Rijke, M., Tjong Kim Sang, E. & Deschacht, K. (2007). The Cornetto Database: Architecture and User-Scenarios. In *DIR*, pp. 89-96.
- Vossen, P., Maks, I., Segers, R. & van der Vliet, H. (2008). Integrating Lexical Units, Synsets, and Ontology in the Cornetto Database. In: *Proceedings on the sixth international Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.