DutchSemCor: building a semantically annotated corpus for Dutch

Piek Vossen[1], Attila Görög[1], Fons Laan[2], Maarten van Gompel[3], Rubén Izquierdo[3], Antal van den Bosch[3]

[1]VU University Amsterdam, De Boelelaan 1105, 1081HV Amsterdam

[2]University of Amsterdam, ...

[3]University of Tilburg, ...

E-mail: p.vossen@let.vu.nl, a.gorog@let.vu.nl, author3@zzz.uuu.edu, author4@hhh.com

Abstract

Most NLP applications require large sense-tagged corpora along with lexical databases to reach satisfactory results in Word Sense Disambiguation (WSD). The number of English language resources for developed WSD increased in the past years, while data scarceness for other languages, however, is more than obvious. The situation is not different for Dutch. In order to overcome this data bottleneck, the DutchSemCor project will deliver a Dutch corpus that is fully sense-tagged with senses from the Cornetto lexical database. Part of this corpus (circa 300K examples) is manually tagged. The remainder will be automatically tagged using different WSD systems and will be validated by human annotators. The project uses existing corpora compiled in other projects but 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.

1. Introduction

Most NLP applications require large sense-tagged corpora along with lexical databases to reach satisfactory results in Word Sense Disambiguation (WSD). While the number of English language resources has increased in the last few years, data is scarce for languages, such as Dutch. In order to overcome the data bottleneck, DutchSemCor[1] is aiming to deliver a one-million word Dutch corpus that is fully 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 provides rich morphosyntactic, semantic and combinatoric information. Synsets in the wordnet part are build-up from 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, circa 300K examples of this corpus 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 parsed and tagged in previous projects. In some cases, the annotators could not find sufficient examples in the corpora for less-frequent word senses. A web-search tool was developed to find additional examples on the Dutch Internet and add these to the data. When writing this paper, 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 for the WSD systems, we annotate the complete corpus for all the senses of Cornetto. In this paper, we will give a description of the project and our approach and we will report on the results so far, both in terms of the manual annotation and the performance of the WSD systems. In section 2, we

[1] http://www2.let.vu.nl/oz/cltl/dutchsemcor/

will 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 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 source that is included in DutchSemCor is the CGN corpus. [HOW MANY TOKENS???], vraag aan Maarten?

Even though SoNaR is large, it still does not contain sufficient examples for certain senses. For this reason, 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[2] , or directly). The results are presented on a screen and human annotators select the samples they want to add to the database. After selection, snippets are automatically tokenised, part-of-speech tagged and lemmatised using Frog[3] and become available in the corpus annotation tool.

The final DutchSemCor corpus will thus be a superset of the SoNaR corpus, the CGN corpus, and the manually-selected snippets. We adapted the corpus representation format FoLiA (Format for Linguistic Annotation[4]) to allow for the annotation of senses, their annotators and their confidence.

The paragraphs in SONAR have been automatically enriched with domain labels from WordNetDomains. For this we used the domain classification software from Irion Technologies that was trained with the words from synsets that belong to a domain in Cornetto. We evaluated the classification on the first part of SONAR, which resulted in an optimal F-measure of 83%. Thanks to the domain classification, tokens in the corpus can be filtered for domains that match certain word meanings.

3. Manual annotation

The envisaged corpus of 1 million tokens was split into two parts that are handled in different ways. The first part of about 300K tokens was annotated manually in a traditional way (See 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 log-file. The log-file is converted into a feature table (Figure-1) by a log-analyzer. The table 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 minimum 80% is reached.

[2] http://www.webcorp.org.uk/

[3] http://ilk.uvt.nl/frog

[4] http://ilk.uvt.nl/folia . FoLiA is based on D-Coi 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.

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 distributior**
11	afschieten	Jonica;Lisanne;	7	2	86	82	100	0.4285714285710	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	afscheiden	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	0.8333333333330	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	0.62666666666668	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	0.8333333333330	0.8533333333333	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	vergooien	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:

dsc-tags2011.09.05 n. Chr. at 1

Figure-1: Logfile converted into feature table

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 which 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, domain labels etc). 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 great syntactic and semantic diversity can better serve WSD techniques and yield better results when used for bootstrapping.

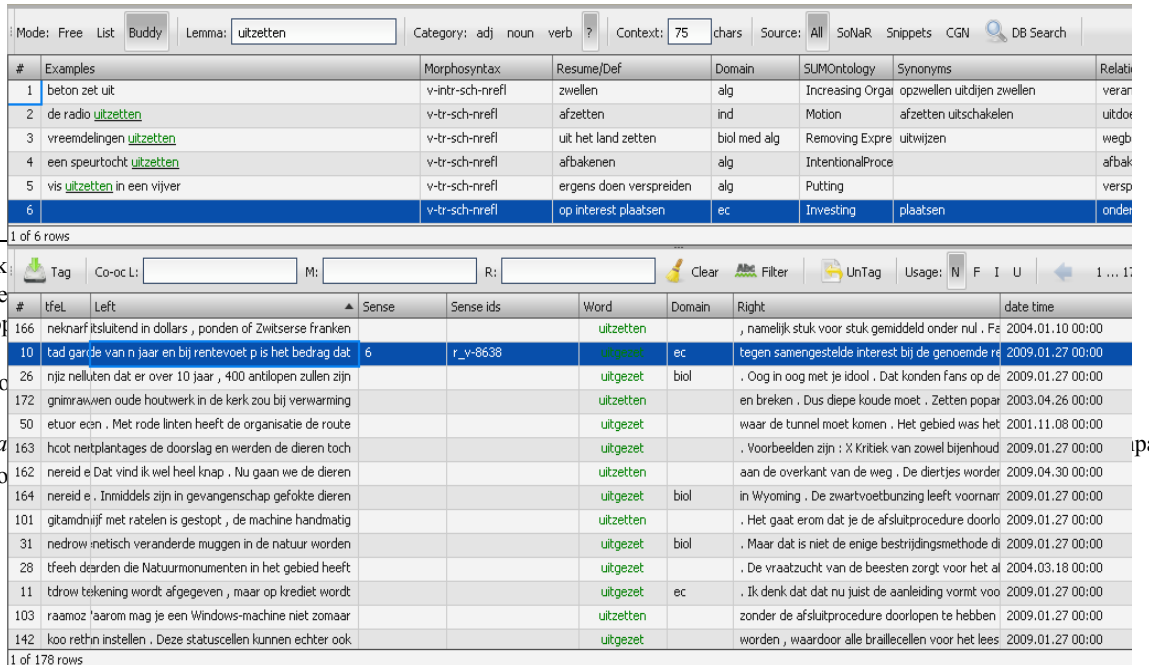
In order to ensure an optimal coherence in the annotation, we have frequent meetings involving the annotation team. In these meetings, we reflect on problems of different origins (possible mistakes in the lexical database, difficult sense distinctions, senses that are not represented in the corpus). Besides, we 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 biweekly for reasons of training and tool-testing. At the moment, 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[5] : 255,625
–
IAA[6] : 93%
–
Coverage 1[7] : 77%
–
Coverage 2[8] : 86%

4. Semantic Annotation Tool (SAT)

The SAT[9] 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 (see 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 corpus lines are reached for each sense.



[5] Tok
[6] Inte
[7] Prop
[8] Pro
[9] The *Ma* panying screensho

Figure-2: SAT interface

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

Figure 3: SAT co-occurrence filtering

5. WSD systems

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

- Knowledge-based WSD that uses the relations in the Cornetto database
- Supervised WSD that creates word-experts from the annotated examples
- Named Entity recognition and Wikification

Named Entity recognition and Wikification is 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 doing WSD.

In this paper, we further discuss the first two approaches. For the knowledge-based WSD we use the UKB-system that was developed by Agirre and Soroa (2009). The UKB considers wordnet as a graph, where the 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. The personalized page-rank algorithm of the UKB then adjusts the weights using the synsets of words that occur in the context of the target word that is disambiguated.

The supervised WSD system uses 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.

In the next sections, we further describe each system and compare their performance.

5.1 WSD-1

The UKB-system requires a lexicon of word forms with pointers to concepts and relations between concepts, from which the graph is built. The Dutch lexicon contains about 84K forms that map to about 70K synsets. Table-1 shows the relations that have been used to build graphs for the UKB. The synset relations (DS:DS) for Dutch are regular wordnet relations as defined in the EuroWordNet project (Vossen 1998). The synset-domain relations (DS:DO) come from WordnetDomains and have been imported through the equivalence relations with the English WordNet. We included the domain hierarchy itself (DO:DO 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 and English wordnet, these are also presented as relations in the UKB (DS:ES). Finally, we have the relations in the English WordNet itself, both the direct relations (ES:ES) as the relations through the disambiguated glosses. (ES:EG). In total, almost 1 million 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 occur in the same sentence. By assuming that these synsets are somehow semantically related, we can derive many new relations for synsets. We extracted two sets: 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.

	Sentences	**Relations**	**Overlap**
Polysemous words	18,653	17,152	2,644
Monosemous words	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 and 3,471 monosemous word relations.

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).[10] 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 *koning* (king), *koningin* (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

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

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

An earlier version of the Dutch UKB1 and UKB3 was evaluated in 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 10th of all participating systems.

5.2 WSD-2

The Supervised WSD system employs k-Nearest Neighbour (Aha et al, 1991) classifiers for word sense disambiguation. This is in line with previous research by Decadt et at (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. A word in this context means a unique lemma, part-of-speech combination. We will first illustrate the workings of the system. Given the corpus and the annotated data gathered by our students, 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 form part of the test set and the remaining 15 (or more if more than 25 examples were collected) are to form part of the training set. All examples that are included in either set have an inter-annotator agreement above a predefined threshold and the minimum number of examples per senses is satisfied for each sense of the word under consideration. 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.

Once the sets have been established, the Supervised WSD system uses this, in addition to the corpus itself, to extract training instances and test instances for each word expert, each instance being one occurrence of the target word, sense annotated by the student, in the corpus. 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, 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 used 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 to systems. 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 occurrence or omission of important predictor words in the same sentence of the sample word is flagged. The global part refers to the fact that whether or not a certain word is an important predictor word for a given target lemma and sense, 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 software Timbl (Daelemans et al., 2007), which is called by the Supervised WSD system to train and test the word-expert classifiers. Timbl may take quite a number of parameters

to tune the classifier performance. The most obvious parameter for k-Nearest Neighbour classification is the value 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 have automated parameter optimisation for Timbl on a per-classifier basis. Thus for each word expert, prior to testing, optimal parameters are sought using leave-one-out cross validation on the training data.

5.3 WSD-results

For the evaluation of the WSD system, we select from the annotated part of the SONAR corpus, all those words with all their senses with at least 25 agreed annotated instances. With all that instances we trainthe word-expert for that word, splitting the instances for each sense into 10 testing and at least 15 training examples

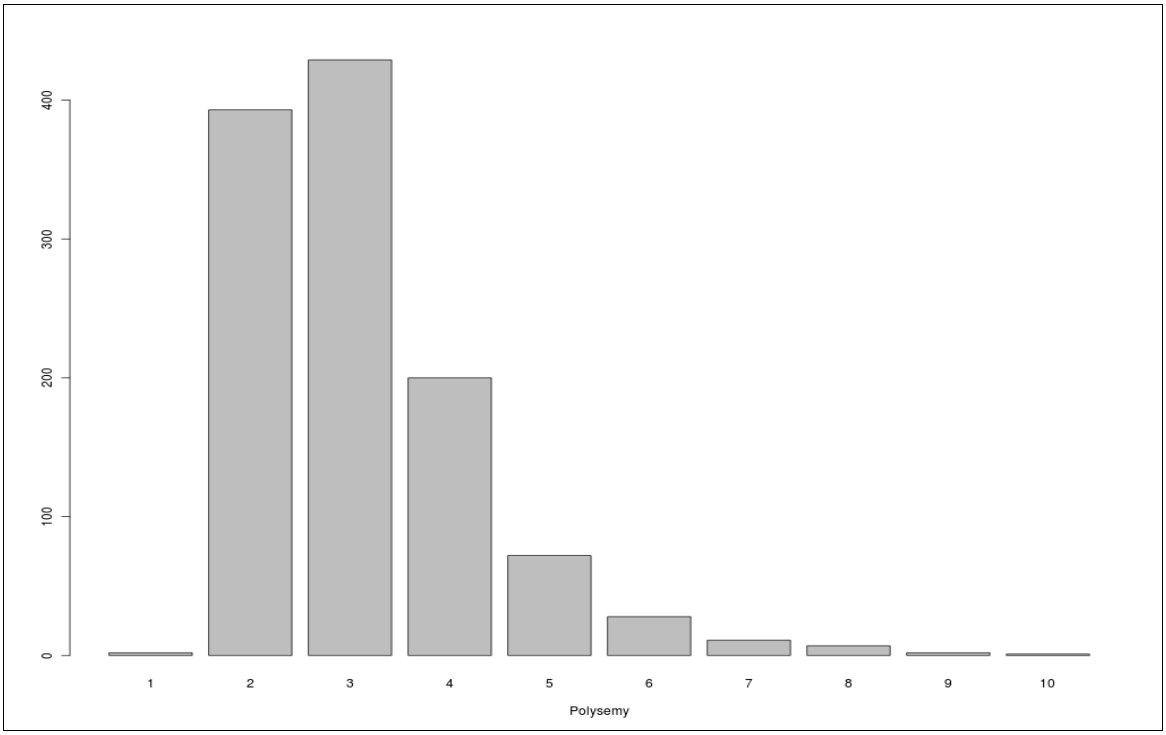
.

The next table shows the performance in terms of token accuracy of the supervised WSD trained with different feature sets, and evaluated over the test set. In this case the training and testing sets were generated from the annotetad part oc SONAR at a early stage of the project, and they only contain 11292 tokens. The size of the context window is shown for each type of feature as subscript. Also two baselines are including, one following a random heuristic, and the other selecting the first sense according to Cornetto.

Feature set	Token accuracy
Chance Baseline	*0.273565*
First sense baseline	*0.276522*
$Words_1$	0.628675
$Words_1$ + $Lemmas_1$	0.634343
$Words_1$ + PoS_1	0.630712
$Words_1$ + $Lemmas_1$ + PoS_1	0.633280
$Words_2$	0.651080
$Words_2$ + $Lemmas_2$	0.648601
$Words_2$ + PoS_2	0.639302
$Words_2$ + $Lemmas_2$ + PoS_2	0.640896
$Words_3$	0.660645
$Words_3$ + $Lemmas_3$	0.653471
$Words_3$ + Bag-of-word	0.721228
$Words_4$	0.655066
$Lemmas_4$	0.657368
$Words_4$ + $Lemmas_4$	0.647538
$Words_5$	0.650283
$Words_6$	0.646652
$Words_7$	0.643819
$Words_8$	0.642490
$Words_9$	0.639479

In general we can see that the effect of consider a wider context does not have a big impact in the performance of the system. The same situation is when enrich the set of features with part-of-speech tags and lemmas. The behaviour is not always as we could expect, and the performance is not higher in all cases when a richer set of features is selected.

We generate another training and testing set in a more advanced stage of the annotacion process, and now the number of tokens is 35338. The polysemy of the testing set can be seen in the next figure. As said before, we do not consider monosemous words in our corpus .The most part of words in the test set have 2 or 3 sense, and then the number of words with more senses is decreased.



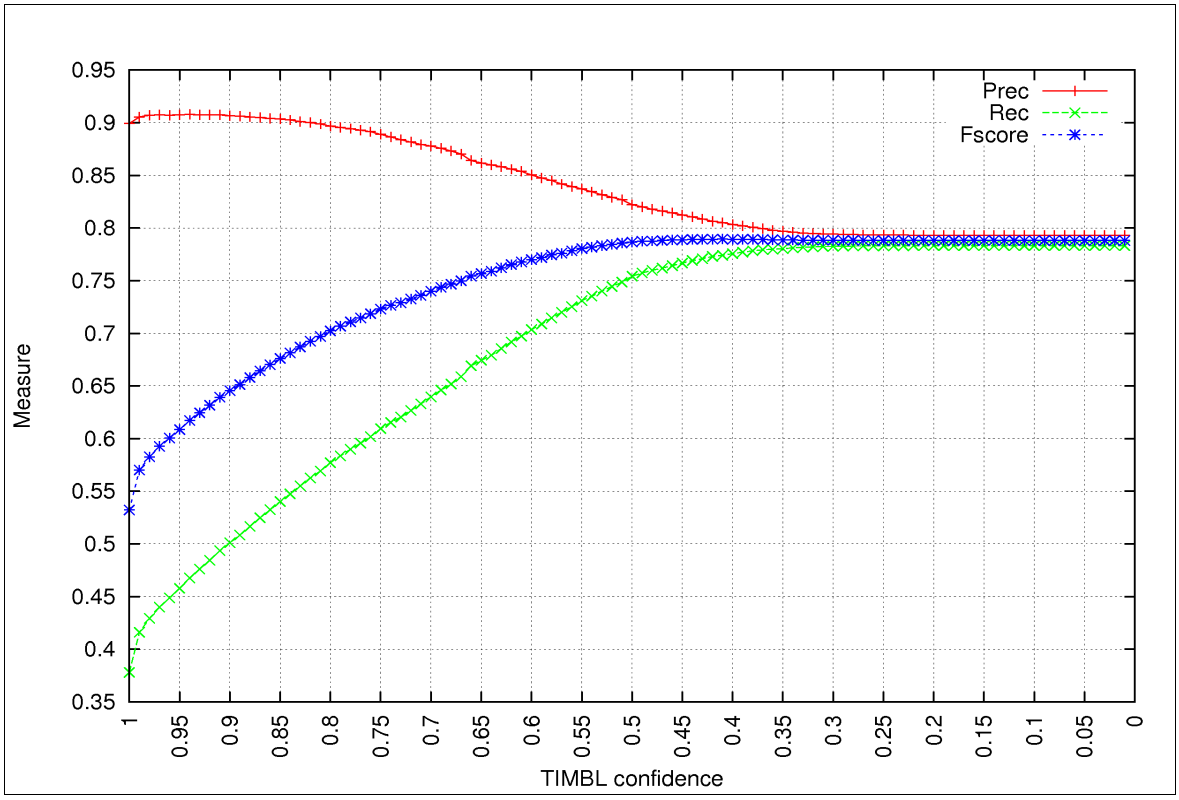
Considering the previous experimentation, we focus now only in some kind of features. Next table shows the performance of the system trained and tested with the new datasets.

Feature set	Token accuracy
$Words_1$	0.646191
$Words_1$ + Bag-of-words	0.725876
$Words_1$ + PoS_1 + Bag-of-words	0.722622
$Words_1$ + Bag-of-words + PS	*0.793084*

As we can see, the use of bag-of-words sets leads to an important improvement of around 8 points in performance. On the contrary, the part-of-speech seems not to help to the classification, not providing any advance. Finally we can see that using the parameter search of TIMBL, the results can be improved in 7 points.

Taking in account the previous results, we select the feature set of the last experiment ($Words_1$ + Bag-of-words + Parameter Search), as the configuration to build our first version of the WSD system. Further experimentation will be carried out using this system.

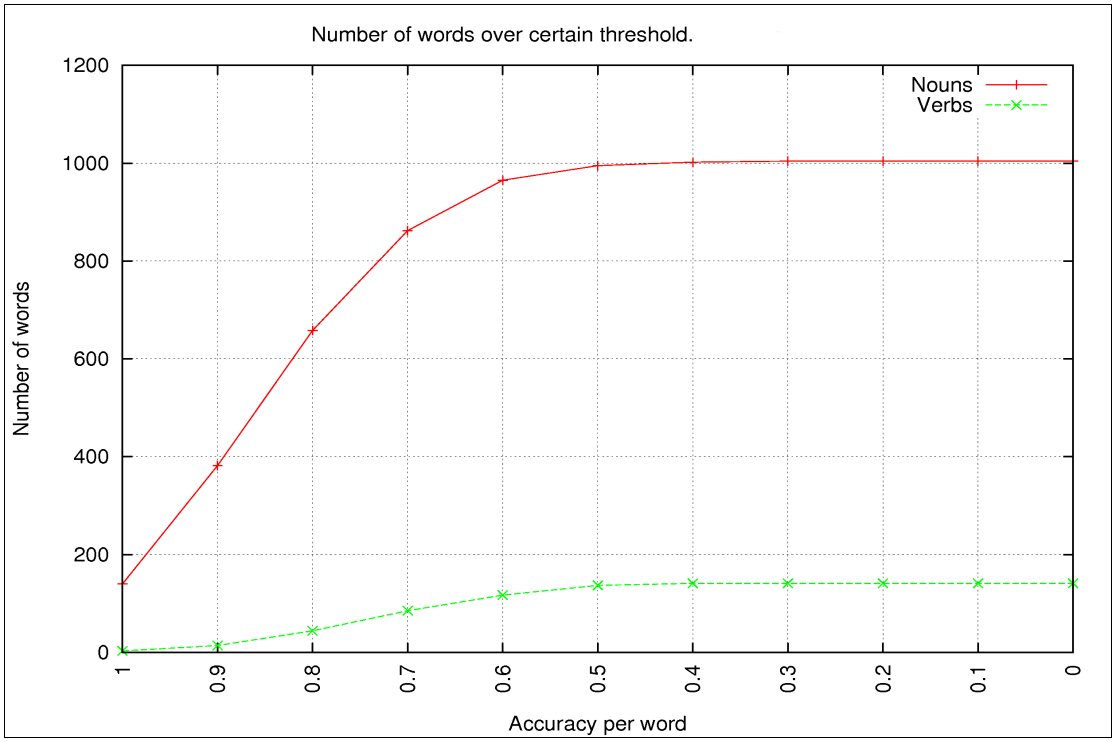
For example in next figure we have performed and analysis of the evolution of our system regarding to the confidence assigned for the TIMBL engine to each token. In the standard evaluation, we consider for each test instance, the sense proposed by TIMBL, regardless the confidence assigned to it. We can make an analysis of how good is that confidence value by filtering out instances with a confidence under a threshold. In this case, the discarded instances remain no tagged, so the recall will be lower, but hopefully the precision will raise.



The behaviour of the system is the expected. When we use a high TIMBL confidence value, the precision is very high, but the recall is hardly penalized. As we choose lower valuyes for the confidence threshold, both values tend to be more similar. It is worth mentioning that selecting a confidence for TIMBL of 0.55, the precision of the system is 0.8370 (+0.439 compared with no filtering) and the Fscore is 0.7804 (only -0.027 less than no filtering). We could filter the test instances according with this threshold of 0.55, improving 4 points the precision and not losing too much recall.

It will be also interesting to know not only the global accuracy of the system, but also how is it performing for individual words. Despite of have a good overall performance, it could be the case that the system is working with high precision for some specific words, but it is reaching low results for other words. In next figure the number of words for which the system reaches a certain accuracy can be seen.

Considering quite high and reasonable mininum accuracy of 0.8, in the case of nouns the 65.54% of words obtain a higher accuracy, and only the 31.21% of verbs overcome this threshold (Remind that the work on verbs will be faced in the second phase of the projects)



6. 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 *Wco*. Words that perform well are ignored.
3. Apply the WSD systems to all occurrences of $w_i \in$ *Wco* that have not been annotated yet.
4. We select the corpus sentences *S* in which the WSD assigned a sense *c* of *wi*, 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 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. UKB can

assign senses to all words in Cornetto. Monosemous words can simply be tagged.

9 Conclusion

10 Acknowledgements

11 References

Aha, D. W. and Kibler, D. and Albert, M. K. (1991). Instance-Based Learning Algorithms. In Journal of Machine Learning, number 1, pages 37-66.

Agirre, E. and A. Soroa. 2009. Personalizing pagerank for word sense disambiguation. In Proc. of EACL-09, pages 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., O. Lopez de Lacalle1, Ch. Fellbaum, S. Hsieh, M. Tesconi4, M. Monachini, P. Vossen, R. Segers (2010) "SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain", in: 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, p. 75-80, Ed. K. Erk & C. Strapparava,, Publ. The Association for Computational Linguistics (ACL), ISBN 978-1-932432-70-1.

Bart Decadt, B., V. Hoste, W. Daelemans, and A. van den Bosch GAMBL, genetic algorithm optimization of memory-based WSD, In: R. Mihalcea and 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.

Daelemans, W., Zravel, J., van der Sloot, K. and 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 proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3), pages 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. and Van Den Bosch, A. (2002). Parameter optimization for machine-learning of word sense disambiguation. *Nat. Lang. Eng.* 8, 4 (December 2002), 311-325.

Kilgarriff, A. (2006). Word senses. In *Word Sense Disambiguation: Algorithms and Applications*. New York, NY : Springer, pp. 29--46.

Magnini, B. and G. Cavaglià. "Integrating Subject Field Codes into WordNet", in Gavrilidou M., Crayannis G., Markantonatu S., Piperidis S. and Stainhaouer G. (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.

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, 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, U.S.A., pp. 1-7.

Ng, H.T. and 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). Association for Computational Linguistics, Stroudsburg, PA, USA, 40-47.

Oostdijk, N. et al. (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, Marocco.

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.

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*, Association for Computational Linguistics, Boulder, Colorado.

Vossen, P. (2006). Cornetto: Een lexicaal-semantische database voor taaltechnologie, *Dixit Special Issue*, Stevin.

Vossen, P. et al. (2007). The Cornetto Database: Architecture and User-Scenarios. In *DIR*.pp.89-96.

Vossen, P. et al. (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, Marocco.