Estimating Supersenses with Conditional Random Fields

Frank Reichartz and Gerhard Paaß

Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
St. Augustin, Germany

Abstract. To interpret unstructured text information on the intranet or internet an interpretation of words and phrases in terms of an ontology is extremely helpful. However many ontologies, e.g. WordNet, are too fine-grained and even human annotators often have disagreements about the precise word sense. Therefore we propose to use coarse-grained supersenses of WordNet which allow to disambiguate most word senses but nevertheless can be assigned with higher reliability. We employ a sequential method for this task, conditional random fields, which allows to take into account the interaction of neighboring words. We use new features as inputs, especially topic models taking into account thematic information in an unsupervised way. With respects to previous results an increase of about 3% in F-value is achieved.

1 Introduction

Most of the information accessible on the internet and intranets is stored as text in different formats. However, although the amount of data available to us is constantly increasing, our ability to absorb this information remains constant. A major obstacle of processing text by computers is its inherent ambiguity: words can have more than one distinct meaning. For example, the 121 most frequent English nouns, which account for about 20% of the words in real text, have on average 7.8 meanings each [AE06]. In spite of this ambiguity humans are nearly unconsciously able to determine the correct word sense.

To interpret and process free text by computer systems the meaning of a word in a sentence has to be determined. *Word sense disambiguation* (WSD) is the process of identifying, which sense of a word is used in a given context. Usually word senses are taken from a given sense inventory, e.g. WordNet [Fel98]. Then WSD is essentially a task of classification: word senses are the classes, the context, i.e. the target word itself and the words in the vicinity of the target word, provides the evidence, and each occurrence of a word is assigned to one or more of its possible classes based on evidence.

One of the major obstacles to high-performance WSD is the fine granularity of available sense inventories. WordNet, for instance, covers more than 100,000 different meanings (synsets). State-of-the-art systems have an accuracy of around 65% in the Senseval-3 all-words task, or 60% [PLDP07] in the SemEval-2007 task. This corresponds to low inter-annotator agreement, e.g. 72.5% agreement

Table 1. 26 Noun Supersenses in WordNet

Supersense	**Nouns denoting ...**	**Supersense**	**Nouns denoting ...**
act	acts or actions	object	natural objects (not man-made)
animal	animals	quantity	quantities and units of measure
artifact	man-made objects	phenomenon	natural phenomena
attribute	attributes of people and objects	plant	plants
body	body parts	possession	possession and transfer of possession
cognition	cognitive processes and contents	process	natural processes
communication	communicative processes and contents	person	people
event	natural events	relation	relations between people or things or ideas
feeling	feelings and emotions	shape	two and three dimensional shapes
food	foods and drinks	state	stable states of affairs
group	groupings of people or objects	substance	substances
location	spatial position	time	time and temporal relations
motive	goals	Tops	abstract terms for unique beginners

in the preparation of the English all-words test set at Senseval-3 [NLH07]. Note that inter-annotator agreement is often considered an upper bound for the performance of WSD systems.

Making WSD an enabling technique for end-to-end applications clearly depends on the ability to deal with reasonable sense distinctions. It is argued that it is not necessary to determine *all* the different senses for every word, but it is sufficient to distinguish the different meanings of a word. Therefore recently a number of researchers investigated word sense disambiguation with a coarse grained sense inventory [CJ03,CA06]. It turned out that in this case it is possible to arrive at inter-annotator agreements of more than 90% [NLH07].

In this paper we follow the approach of [CA06] in using WordNet supersenses for nouns and verbs. As the meaning of a word is mainly determined by the sequence of words in the vicinity we apply conditional random fields as sequence modelling technique. It is able to take into account an enormous number of features and propagate evidence between words. In addition we use topic modelling as a new feature which can collect evidence about the meaning of a word in an unsupervised way. The resulting level of accuracy is higher than for previous approaches.

Table 2. 15 Verb Supersenses in WordNet

Supersense	**Verbs denoting ...**	**Supersense**	**Verbs denoting ...**
body	grooming, dressing and bodily care	emotion	feeling
change	size, temperature change, intensifying	motion	walking, flying, swimming
cognition	thinking, judging, analyzing, doubting	perception	seeing, hearing, feeling
communication	telling, asking, ordering, singing	possession	buying, selling, owning
competition	fighting, athletic activities	social	political and social activities and events
consumption	eating and drinking	stative	being, having, spatial relations
contact	touching, hitting, tying, digging	weather	raining, snowing, thawing, thundering
creation	sewing, baking, painting, performing		

Supersense tagging has many potential applications. It has been shown [CHJ03] that supersense information can support supervised WSD, by providing a partial disambiguation step. Together with other sources of information such as part-of-speech tags, domain-specific extracted named entities, chunks or shallow parses the information generated by supersense tagging may be useful in tasks such as question answering and semantic information extraction and retrieval, where large amounts of text need to be processed.

In the second section we describe the supersense inventory of WordNet in an example. The next section describes learning approaches to supersense tagging used up to now. Subsequently we sketch conditional random fields and their application to supersense tagging. The fifth section lists the features we use in our model, especially the scores of an unsupervised topic model. The next section describes the corpora used for our experiments and the results. The final section gives a summary of the paper.

2 WordNet Senses and Supersenses

WordNet [Fel98] is a semantic lexicon for the English language. It groups English words into sets of synonyms called *synsets*, provides short definitions, and contains various semantic relations between synonym sets. It includes 11,306 verbs mapped to 13,508 word senses, called synsets, and 114,648 common and proper nouns mapped to 79,689 synsets.

Each noun or verb synset is associated with one of 41 broad semantic categories called supersenses [CJ03]. There are 26 supersenses for nouns described in table 1 and 15 for verbs shown in table 2. In the WordNet graphical user interface the supersenses can be seen by checking the option "Lexical File Information".

Table 3. Synsets of Noun "blow"

Synset	**Definition and Example**	**Supersense**
blow	a powerful stroke with the fist or a weapon; "a blow on the head"	noun.act
blow, bump	an impact (as from a collision); "the bump threw him off the bicycle"	noun.event
reverse, reversal, setback, blow, black eye	an unfortunate happening that hinders or impedes; something that is thwarting or frustrating	noun.event
shock, blow	an unpleasant or disappointing surprise; "it came as a shock to learn that he was injured"	noun.event
gust, blast, blow	a strong current of air; "the tree was bent almost double by the gust"	noun.phenomenon
coke, blow, nose candy, snow, C	street names for cocaine	noun.artifact
blow, puff	forceful exhalation through the nose or mouth; "he gave his nose a loud blow"	noun.act

This coarse-grained supersense inventory has a number of attractive features for the purpose of annotating natural language meanings. First, the small size of the set makes it possible to build a single model which has positive consequences on robustness. Second, classes, although fairly general, are easily recognizable and not too abstract or vague. More importantly, similar word senses tend to be merged together.

As an example, table 3 summarizes all senses of the noun "blow". The rows in the table are ordered according to the frequency of the corresponding synset. The 7 synsets of "blow" are mapped to 4 supersenses: act, event, phenomenon, and artifact. The second, third and forth sense are quite similar and are merged in the "event" supersense removing small sense distinctions which are hard to discriminate. The remaining four senses of "blow" are discriminated by their supersenses. Hence in this case the most important sense distinctions made by WordNet are kept at the supersense level. Therefore the assignment of supersenses at least partially discriminates between different synsets. Note that the most common subsense of "blow" has a different supersense then the other synsets; however, this is not always the case. There are 22 synsets of the verb blow in table 4. Many of these synsets are quite hard to distinguish. As, however, 9 of the 10 different supersenses for verbs are covered, even the small number of supersenses discriminate between many different aspects of "blow".

WordNet also includes a limited number of named entities, e.g. "Planck", "Max Planck" and "Max Karl Ernst Ludwig Planck" define a synset with supersense noun.person. These terms can be detected by named entity recognition approaches. Hence for the usual named entity recognition categories person, group, location, time, and artifacts (e.g. products) we might distinguish between proper nouns and common nouns.

Table 4. Synsets of Verb "blow"

Synset	**Definition**	**Supersense**
blow	exhale hard	verb.body
blow	be blowing or storming	verb.weather
blow	free of obstruction by blowing air through	verb.body
float, drift, be adrift, blow	be in motion due to some air or water current	verb.motion
blow	make a sound as if blown	verb.perception
blow	shape by blowing	verb.change
botch, bodge, bumble, fumble, ...	make a mess of, destroy or ruin	verb.social
waste, blow, squander	spend thoughtlessly; throw away	verb.possession
blow	spend lavishly or wastefully on	verb.possession
blow	sound by having air expelled through a tube	verb.perception
blow	play or sound a wind instrument	verb.perception
fellate, blow, go down on	provide sexual gratification through oral stimulation	verb.perception
blow	cause air to go in, on, or through	verb.motion
blow	cause to move by means of an air current	verb.motion
blow	spout moist air from the blowhole	verb.motion
shove off, shove along, blow	leave; informal or rude	verb.motion
blow	lay eggs	verb.contact
blow	cause to be revealed and jeopardized	verb.communication
boast, tout, swash, shoot a line, brag,...	show off	verb.communication
blow	allow to regain its breath	verb.communication
blow out, burn out, blow	melt, break, or become otherwise unusable	verb.change
blow	burst suddenly	verb.change

There are two ways to find the supersense of a synset. The simple way, which is also followed in this paper, is to use the unique lexicographical category assigned to the synset by the WordNet annotators. The alternative way is to exploit the hypernym relationship. In this way we may find all synsets corresponding to supersenses which are related to the target synset by a hypernym path. Occasionally there exist several supersenses related to a single target synset in this way. The exploitation of these multiple labels will be left to future work.

The learning task of supersense tagging is the assignment of the correct supersense to the target word. The annotation model is trained using a corpus of sentences annotated with the correct supersense. A prominent corpus annotated with WordNet senses is SemCor described below. Usually the annotation is done simultaneously for all nouns and verbs in the corpus. It is clear that the

supersenses of neighboring words have a strong relation to the supersense of the target word. This should be taken into account by the annotation method.

3 Learning Approaches for Supersense Tagging

While fine grained word disambiguation has to cope with thousands of categories the coarse grained word sense disambiguation task only relatively few classes have to be taken into account. This leads to the utilization of specific machine learning methods. A first approach is based on *classification methods*, which for a target word x get a description of the neighborhood $N(x)$ as input and estimate the corresponding supersense y as output class.

$$y = f(N(x)) \tag{1}$$

This usually implies that each word is labelled individually without taking into account interactions. The bag-of-word representations of input features used for document classification is insufficient in this case, as the relative distance of a word or feature to the target word x is important. Therefore it is necessary to encode the words and derived features as well as their relative position to the target word x. The neighborhood $N(x)$ usually includes local collocations, parts-of-speech tags, and surrounding words. Examples of classifiers are multiclass averaged perceptrons [CJ03], the naive Bayes classifier [CLT07], and the support vector machine [CNZ07], where the last two papers describe positive results at the SemEval07 competition.

A second approach to coarse-grained supersense tagging relies on *sequential models*, which are common in NER, POS tagging, shallow parsing, etc.. Although it seems reasonable to assume that occurrences of word senses in a sentence can be correlated, hence that structured learning methods could be successful, there has not been much work on sequential WSD. Probably the first to apply a Hidden Markov Model tagger to semantic disambiguation were [SSGC97]. To make the method more tractable, they also used a supersense tagset and estimated the model on SemCor. By cross-validation they show a marked improvement over the first sense baseline.

A more elaborate sequential model is used by [CA06]. They tackle the problem of assigning WordNet supersenses to nouns and verbs and use a discriminatively trained Hidden Markov Model, which was proposed by [Col02]. These models have several advantages over generative models, such as not requiring questionable independence assumptions, optimizing the conditional likelihood directly and employing richer feature representations. Overall their supersense tagger achieves F-scores between 70.5 and 77.2%.

[DM06] target fine grained WSD for the WordNet synsets using a Conditional Random Field (CRF). This model allows to take into account a variety of non-independent input features. They adapt the CRF to the large number of labels to be predicted by taking into account the hypernym/hyponym relation and report a marked reduction in training time with only a limited loss in accuracy.

4 Conditional Random Fields

In this paper we show that by using new features we can improve the performance of sequential models. We consider a task which is characterized by sequences $\mathbf{x} = (x_1, \ldots, x_T)$ of inputs. In language modelling an input x_t usually contains different features of the t-th words of a document $\mathbf{x}$. To each word x_t corresponds a state y_t which has values in a set of labels $\mathcal{Y} = \{\gamma_1, \ldots, \gamma_m\}$, e.g. supersenses. It is the task to predict the state sequence $\mathbf{y} = (y_1, \ldots, y_T)$.

Conditional Random fields (CRFs) [LMP01,SM07] are conditional probability distributions that factorize according to an undirected model.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \phi_c(\mathbf{y}_c, \mathbf{x})$$

Here $\mathbf{y}_c = (y_t)_{t \in I_c}$ is a subvector of states $\mathbf{y} = (y_1, \ldots, y_T)$ with indices $t \in I_c \subseteq \{1, \ldots, T\}$. The $\phi_c(\mathbf{y}_c, \mathbf{x})$ are real-valued functions of these variables and $Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}^T} \prod_{c \in C} \phi_c(\mathbf{y}_c, \mathbf{x})$ is a normalizing factor for the sequence $\mathbf{x}$.

Many applications use a linear-chain CRF, in which the sequential order of inputs is taken into account and used for a first-order Markov assumption on the dependence structure. In this case the subvectors are pairs $\mathbf{y}_c = (y_t, y_{t-1})$ and yield the *feature functions* $f_k(y_t, y_{t-1}, \mathbf{x})$ with an associated parameter λ_k. We assume that the corresponding feature functions do not depend on the value of t, which allows weight sharing between all these components. On the other hand they may take into account the complete input vector $\mathbf{x}$. In the simplest case feature functions take the value 1 for a subset of the values $y_t, y_{t-1}, \mathbf{x}$ and 0 otherwise. Note that there may be different feature functions for the same variables $y_t, y_{t-1}, \mathbf{x}$. This also covers the special case of functions $g_k(y_t, \mathbf{x})$ containing only one state and can easily be extended to higher order Markov chains.

If $r(\mathbf{x})$ is a component only depending on $\mathbf{x}$ we may write the joint distribution as

$$p(\mathbf{y}, \mathbf{x}) = \exp\left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right) r(\mathbf{x}) \tag{2}$$

Then the conditional distribution is

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y} \in \mathcal{Y}^T} p(\mathbf{y}, \mathbf{x})} = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right) \tag{3}$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}^T} \exp\left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right)$ is an input-specific normalization function.

Now assume we have N observations $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$. As a regularizer we introduce a penalty for large λ-values, e.g. a Gaussian prior $\exp(-\sum_{k=1}^{K} \lambda_k^2 / 2\sigma^2)$. Then the conditional log-likelihood $L(\lambda) = \sum_{n=1}^{N} \log p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \lambda)$ for the vector λ of all parameters is

$$L(\theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(n)}, y_{t-1}^{(n)}, \mathbf{x}^{(n)}) \\ - \sum_{n=1}^{N} log \sum_{y \in \mathcal{Y}^T} exp \left(\sum_{t} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(n)}, y_{t-1}^{(n)}, \mathbf{x}^{(n)}) \right) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

The derivative of the log-likelihood may be evaluated and used by conjugate gradient maximizers to find the optimal parameters.

5 Features Used for Supersense Tagging

In principle the whole input sequence $x = (x_1, \ldots, x_t, \ldots, x_T)$ and all derived features may be used in the feature functions $f_k(y_t, y_{t-1}, x)$. We used only features of words in the neighborhood of the target word, where a neighborhood covers the preceding 2 words the word itself and the subsequent two words:

1. `Word`: The target word x_t itself.
2. `Lemma`: The lemma for the previous word x_{t-1}.
3. `Prefix`: Three-character prefix of the target word x_t.
4. `Suffix`: Three-character suffix of the target word x_t.
5. `Potential Supersenses` with lags: $pot(x_{t-2}), pot(x_{t-1}), pot(x_t), pot(x_{t+1}), pot(x_{t+2})$ where $pot(w)$ is a function mapping the word w to all potential supersenses taking into account the polysemy of w. For the noun blow, for instance, we would have 5 potential supersenses according to table 3.
6. `Part-of-Speech tags` as used for the Brown corpus (82 different tags) with lags: $pos(x_{t-2}), pos(x_{t-1}), pos(x_t), pos(x_{t+1}), pos(x_{t+2})$ where $pos(x)$ is the POS-tag associated with the word x.
7. `Coarse part-of-speech tags` containing only the first character of the tags with lags: $POS(x_{t-2}), POS(x_{t-1}), POS(x_t), POS(x_{t+1}), POS(x_{t+2})$.
8. `Word Shape` : Features based on the capitalization and other shape characteristics of a word w. Especially we use $icap(x_{t-1}), icap(x_t), icap(x_{t+1})$ where $icap(x)$ is the function which indicates if the initial letter of x is capitalized. In addition we employ the features $mcap(x_{t-1}), mcap(x_t), mcap(x_{t+1})$ where $mcap(x)$ indicates whatever the word x contains mixed capitalization or not. The last shape features are $acap(x_{t-1}), acap(x_t), acap(x_{t+1})$ where $acap(x)$ is 1 if w is completely capitalized.
9. `Topic-Model`: We use a Latent Dirichlet Allocation (LDA) topic model with 50 different topics. The algorithm assigns to each word the probability that the word belongs to the topic. We define three threshold values 0.1, 0.8, and 0.98. If for word x_t the probability of topic i is above 0.1 then feature $top_{i,0.1}(x_t)$ is set to 1. Correspondingly $top_{i,0.8}(x_t)$ and $top_{i,0.98}(x_t)$ are set to 1 if the probability of topic i is above 0.8 or 0.98 respectively. Therefore we get 150 possible features for each word x_t.

Table 5. Results for Noun Supersenses

	Frequency					
Supersense	**potential**	**true**	**Precision**	**Recall**	**F1**	**stdErr F1**
act	23493	7962	0.756	0.760	0.758	0.0070
animal	3449	1028	0.868	0.847	0.857	0.0083
artifact	25288	8894	0.827	0.843	0.835	0.0026
attribute	15835	4719	0.709	0.719	0.714	0.0038
body	12369	2555	0.872	0.890	0.881	0.0068
cognition	22302	7018	0.737	0.740	0.738	0.0030
communication	27817	6952	0.788	0.779	0.784	0.0017
event	7360	1806	0.636	0.618	0.627	0.0129
feeling	2418	831	0.727	0.712	0.719	0.0089
food	2769	600	0.897	0.824	0.859	0.0120
group	13919	4854	0.809	0.817	0.813	0.0061
location	10311	3558	0.800	0.813	0.806	0.0072
motive	728	133	0.674	0.617	0.640	0.0274
object	5671	1484	0.765	0.723	0.743	0.0114
person	17015	8172	0.946	0.943	0.944	0.0023
phenomenon	4256	1190	0.750	0.740	0.745	0.0126
plant	1572	447	0.891	0.852	0.870	0.0159
possession	4961	1483	0.791	0.780	0.785	0.0106
process	2299	501	0.744	0.693	0.717	0.0131
quantity	6653	1906	0.862	0.844	0.853	0.0079
relation	3069	913	0.701	0.719	0.709	0.0123
shape	2246	333	0.645	0.536	0.583	0.0339
state	13968	3380	0.705	0.686	0.695	0.0060
substance	5407	2079	0.846	0.855	0.850	0.0045
time	7200	4158	0.901	0.910	0.906	0.0034
Tops	12127	9785	0.977	0.979	0.978	0.0007

The number of features has to be restricted to avoid overfitting. In future versions we will use statistical feature selection.

The LDA model was developed by [BNJ03] as a generative probabilistic model for text data. It is an extension of the latent semantic indexing model. Each topic defines a distribution over the words in a collection and each document is defined as a mixture of topics. Based on the context LDA groups words often occurring together into soft clusters. As it takes into account the context of words it is able to distinguish between different meanings of a word. We use an efficient version employing mean field approximation for fast estimation.

Table 6. Results for Verb Supersenses

	Frequency					
Supersense	**potential**	**true**	**Precision**	**Recall**	**F1**	**stdErr F1**
body	8377	822	0.686	0.629	0.656	0.0076
change	15230	4086	0.693	0.687	0.690	0.0109
cognition	14963	4253	0.769	0.762	0.765	0.0041
communication	18907	6336	0.811	0.804	0.808	0.0060
competition	6164	549	0.591	0.516	0.550	0.0281
consumption	5838	825	0.793	0.767	0.779	0.0106
contact	13492	2693	0.685	0.683	0.684	0.0073
creation	10344	1689	0.578	0.529	0.552	0.0124
emotion	3960	1038	0.769	0.766	0.767	0.0138
motion	12413	3777	0.730	0.763	0.746	0.0029
perception	9653	2597	0.725	0.728	0.726	0.0075
possession	19553	3031	0.653	0.701	0.676	0.0041
social	24664	3535	0.669	0.656	0.662	0.0095
stative	25148	12181	0.897	0.904	0.901	0.0027
weather	428	57	0.733	0.343	0.435	0.0867

Table 7. Results

Method	**Precision**	**Recall**	**F1**	**stdErr**
CRF	0.803	0.802	0.802	0.002
Ciramita(CA06)	0.766	0.777	0.771	0.004
Baseline(First-Sense)	0.639	0.692	0.664	

6 Experiments

6.1 Corpora

In the SemEval Coarse Grained task [NLH07] specific supersenses are constructed for annotation using the Oxford Dictionary of English. To be compatible with the results of [CA06] we instead used the original WordNet supersenses as targets and SemCor as training corpus. The SemCor dataset [MLTB93] consists of documents taken from the Brown Corpus. It is freely available[1] and contains 352 documents which were manually annotated. SemCor can be broken down into two subsets. The first subset `SemcorN` contains the 186 documents where all open class words (nouns, adjectives, verbs and adverbs) are annotated with Part-of-Speech (POS), lemma and the correct WordNet synset. In the second subset `SemcorV` of 166 documents only the verbs are annotated. We only used the first subset `SemcorN` for our experiments.

[1] http://www.cs.unt.edu/ rada/downloads.html

6.2 Evaluation

As stated at the outset a supersense is assigned to each noun and verb. Multiword phrases such as person names were treated as one word in this experiment. Therefore we just evaluated if the correct supersense was assigned to a noun or verb and used the usual evaluation measures of precision, recall and F1.

Although most words have several supersenses there is usually one sense which is correct in most cases. A quite successful strategy is to assign the most frequent supersense to a word. As a comparison we also provide the most frequent sense to show the improvement over this simple strategy.

6.3 Results

Our experiments where conducted on a cluster of 10 machines each equipped with two 2.8GHz Intel Dualcore processors. We parallelized the original CRF-implementation of Mallet [McC02] to make full use of the 40 CPUs available in our cluster. One training run took an average of 300 iterations and a runtime of roughly 10 hours. The parallelization reduced the training time by about 90%.

In table 5 the results for noun supersenses are shown based on five-fold crossvalidation. The third column shows how often the supersense occurs in the collection. The second column shows how often the supersense is a potential supersense of the word and indicates the ambiguity of the term. The F1-value is between 0.58 and 0.98 with a frequency-weighted average of 0.82. Table 6 contains the results for verb supersenses. Here the F1-value is somewhat lower with values between 0.43 and 0.90 with a frequency-weighted average of 0.765. Due to high ambiguity supersenses of verbs are much more difficult to determine.

In table 7 the frequency-weighted averages are shown. Here our CRF solution yields an F1-value of 0.802 with standard error of 0.002. This approximate standard error is determined from the standard deviation of the 5 crossvalidation results. Note that there are no systematic differences between precision and recall. Compared to the value achieved by [CA06] we get an increase of 0.031, which corresponds to a 13% error reduction. If we consider the most frequent sense baseline we have a marked increase of F1 of about 0.14.

7 Summary

We used conditional random fields to model the sequential context of words and their relation to supersenses. In addition we included new features into the model. Compared to previous results we were able reduce the F1-value by 3.1% corresponding to a reduction of error for 13%. As the annotation time for a new document is quite low the approach may be used to annotate large collections of documents in spite of the high training time. Future research aims at including larger corpora into the training process to reduce the error even further.

References

[AE06] Eneko Agirre and Phillip Edmonds. Introduction. In *Word Sense Disambiguation: Algorithms and Applications*, pages 1–28. Springer, 2006.

[BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[CA06] M. Ciaramita and Y. Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[CHJ03] M. Ciaramita, T. Hofmann, and M. Johnson. Hierarchical semantic classification: Word sense disambiguation with world knowledge. In *Proceedings of IJCAI 2003*, 2003.

[CJ03] Massimiliano Ciaramita and Mark Johnson. Supersense tagging of unknown nouns in wordnet. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, pages 168–173, 2003.

[CLT07] Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. Nus-ml: Improving word sense disambiguation using topic features. In *Proc. SemEval-2007*, pages 249–252, 2007.

[CNZ07] Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proc. SemEval-2007*, pages 253–256, 2007.

[Col02] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8, 2002.

[DM06] Koen Deschacht and Marie-Francine Moens. Efficient hierarchical entity classifier using conditional random fields. In *Proc. 2nd Workshop on Ontology Learning and Population*, pages 33–40, 2006.

[Fel98] Christine Fellbaum. *WordNet: An Electronic Lexical database.* MIT Press, 1998.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[McC02] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[MLTB93] George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. A semantic concordance. In *HLT '93: Proc. workshop on Human Language Technology*, pages 303–308, 1993.

[NLH07] Roberto Navigli, Kenneth Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proc. Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, 2007.

[PLDP07] Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proc. Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, 2007.

[SM07] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT-Press, 2007.

[SSGC97] F. Segond, A. Schiller, G. Grefenstette, and J.P. Chanod. An experiment in semantic tagging using hidden markov model. In *Proc. Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources (ACL/EACL 1997)*, pages 78–81, 1997.