Exploiting Semantic Constraints for Estimating Supersenses with CRFs*

Gerhard Paaß† Frank Reichartz‡

Abstract

The annotation of words and phrases by ontology concepts is extremely helpful for semantic interpretation. However many ontologies, e.g. WordNet, are too fine-grained and even human annotators often have disagreements about the precise word sense. Therefore we use coarse-grained supersenses of WordNet. We employ conditional random fields (CRFs) to predict these supersenses taking into account the interaction of neigboring words. As the annotation of training data is costly we modify the CRF algorithm to process lumped labels, i.e. a set of possible labels for each training example, one of which is the correct label. This information can be derived automatically from WordNet. By this and the use of new features we are able to increase the f-value for about 8% compared to previous results. If we use training data without annotations it turns out that the resulting F-value is only slightly lower than for a fully labeled training data. Therefore unlabeled data may be employed in an unsupervised way to increase the quality of predicted supersenses.

1 Introduction

Most of the information accessible in the internet and intranets is stored as text in different formats. A major obstacle of using this information by computers is the inherent polysemy: 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 [1]. In spite of this ambiguity humans are nearly unconciously able to determine the correct word sense.

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 [12]. Then WSD is essentially a task of classification: word senses are the classes, the context, i.e. the words in the vincinity of the target word, provides the evidence, and each occurence of a word is assigned to one or more of its possible classes based on evidence.

One of the major difficulties for 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% [21] in the SemEval-2007 task. This corresponds to low inter-annotator agreement, e.g. 72.5% agreement in the preparation of the English all-words test set at Senseval-3 [19]. 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 coarse meanings of a word [13]. Therefore recently a number of researchers investigated word sense disambiguation with a coarse grained sense inventory [9] [19]. It turned out that in this case it is possible to arrive at inter-annotator aggreements of more than 90% [19].

In this paper we follow the approach of [7] in using supersenses for nouns and verbs. As the meaning of a word is mainly determined by the sequence of words in the vincinity we use the conditional random field (CRF) [15] as sequence modelling technique. It is able to take into account an enormous number of context features and to propagate evidence between words. In addition we use topic modelling [3] as a new feature which can collect evidence about the meaning of a word in an unsupervised way.

Many words in WordNet may have only a few different supersenses. We exploit this information by applying the CRF to non-annotated text. As supersense training labels for a word we use the set of all possible supersenses, which can be considered as a *lumped label*. The CRF in this way may learn that a large number of possible transitions between states are unlikely. In the paper the CRF training algorithm is adapted to this new type of training data. Empirical experiments indicate the effectiveness of the approach.

Supersense tagging has many potential applications. It has been shown [8] that supersense information can sup-

*Supported by the Theseus research program of the German Federal Ministry of Economics and Technology.

†Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Sankt Augustin, Germany. *gerhard.paass@iais.fraunhofer.de*

‡Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Sankt Augustin, Germany. *frank.reichartz@iais.fraunhofer.de*

Table 1: 26 Noun Supersenses in WordNet

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

Table 2: 15 Verb Supersenses in WordNet

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

port 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 is useful in tasks such as question answering as well as 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 following section describes the gradient calculations for the new lumped label CRF. The sixth 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 [12] 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* [9]. 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".

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 fourth 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 supersense different from the other synsets; however, this is not always the

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

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. This would be important for a real-world application as named entities often consist of several words and are usually formed in different ways than common nouns.

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

Fine grained word disambiguation has to cope with thousands of categories. For the coarse grained word sense disambiguation task, however, 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.

$$y = f(N(x)) \tag{3.1}$$

This usually implies that each word is labelled individually without taking into account interactions between supersenses. The bag-of-word representations of input features used for document classifications 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 [9], the naive Bayes classifier [5], and the support vector machine [6]. The last two approaches yield good results at the SemEval07 competition.

[20] consider the case that only lumped labels are available for classification, i.e. for each training example a set of possible labels is observed, one of which is the correct label. They formulate a convex quadratic optimization problem using hinge loss. Experiments with different datasets show that the utilization of partial labels improves the performance of classification and yields reasonable performance in the absence of any fully labeled data.

A second approach to coarse-grained supersense tagging relies on *sequential models*, which are common in Named-Entity-Recognition, Part-of-Speech tagging, shallow parsing, etc.. Sequential models potentially have a higher reliability as they can take into account the labels of neighboring

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

words during prediction, which is not directly possible for classification methods. Although it is reasonable to assume that occurrences of word senses in a sentence can be correlated, there has not been much work on sequential WSD. Probably the first to apply a Hidden Markov Model tagger to semantic disambiguation were [23]. To make the method more tractable, they also used a supersense tagset and estimated the model on the Semcor corpus. By cross-validation they show a marked improvement over the first sense baseline.

A more elaborate sequential model is used by [7]. They tackle the problem of assigning WordNet supersenses to nouns and verbs and use a discrimiatively trained Hidden Markov Model, which was proposed by [10]. 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%.

Deschacht and Moens [11] 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 use the CRF for supersense tagging. 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) [15, 25] are conditional probability distributions that factorize according to an undirected model.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \phi_c(\mathbf{y}_c, \mathbf{x}) \tag{4.2}$$

It is assumed that there is a dependency between the states at different sequence positions. The $c \in C$ are subsets $c \subseteq \{1, \ldots, T\}$ of time instances where the states may have direct interactions. $\mathbf{y}_c = (y_t)_{t \in c}$ is a subvector of states $\mathbf{y} = (y_1, \ldots, y_T)$ with indices $t \in c \subseteq \{1, \ldots, T\}$. 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. Hence we arrive at

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \right) \tag{4.3}$$

If let $p(\mathbf{x})$ be the unknown marginal distribution of $\mathbf{x}$. As $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}, \mathbf{x})/p(\mathbf{x})$ we may write the joint distribution by (4.3) as

$$p(\mathbf{y}, \mathbf{x}) = r(\mathbf{x}) \exp \left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \right) \tag{4.4}$$

where $r(\mathbf{x}) = p(\mathbf{x})/Z(\mathbf{x})$ is a term depending only on $\mathbf{x}$. Then the conditional distribution is

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{p(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y} \in \mathcal{Y}^T} p(\mathbf{y}, \mathbf{x})} \\ &= \frac{1}{\tilde{Z}(\mathbf{x})} \exp \left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \right) \end{aligned} \tag{4.5}$$

where $\tilde{Z}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}^T} \exp \left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}) \right)$ is an input-specific normalization function and $\mathcal{Y}^T$ is the set of all sequences of the form $\mathbf{y} = (y_1, \ldots, y_T)$.

Now assume we have N i.i.d. observations $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$, $n = 1, \ldots, N$ generated according to (4.4). As a regularizer we introduce a penalty for large λ-values, e.g. a prior $p(\lambda) = \exp(-\sum_{k=1}^{K} \lambda_k^2/2\sigma^2)$ proportional to a Gaussian. Then the conditional log-likelihood $L(\lambda)$ for the vector λ of all parameters is

$$\begin{aligned} L(\lambda) &= \sum_{n=1}^{N} \log p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \lambda) \\ &= \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(n)}, y_{t-1}^{(n)}, \mathbf{x}^{(n)}) \\ &\quad - \sum_{n=1}^{N} \log \sum_{y \in \mathcal{Y}^T} \exp \left(\sum_t \sum_{k=1}^{K} \lambda_k f_k(y_t^{(n)}, y_{t-1}^{(n)}, \mathbf{x}^{(n)}) \right) \\ &\quad - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \end{aligned} \tag{4.6}$$

The derivative of the log-likelihood may be evaluated and used by limited memory quasi-Newton maximizers like L-BFGS [16] to find the optimal parameters.

5 Likelihood for Lumped Labels

5.1 Likelihood Value In the training set for conventional CRFs the exact labels have to be known. In some cases, however, a set of possible labels can be inferred. This holds, for instance, in supersense tagging, where a word only has a few possible senses and hence only a few possible supersenses. As these potential supersenses may be derived automatically we may generate training data without human tagging effort. To our knowledge the likelihood and gradient calculations presented in this section have not been presented previously.

Assume instead of observing a single value γ_j for y_t we only know that y_t is contained in some nonempty subset $Y_t \subseteq \mathcal{Y}$ of states. In the case of a singleton $Y_t = \{\gamma_j\}$ we have a single observation as usual, while the case $Y_t = \mathcal{Y}$ corresponds to completely missing information. Note that no assumption is used about the relative probability of states in Y_t. The i-th observation sequence now has the form of a cross product

$$Y^{(i)} = Y_1^{(i)} \times \cdots \times Y_{T_i}^{(i)}$$

The likelihood of all observations has to take into account the probability of all output sequences $\prod_{i=1}^{N} p(Y^{(i)}|\mathbf{x}^{(i)}, \lambda) = \prod_{i=1}^{N} p(\mathbf{y}^{(i)} \in Y_1^{(i)} \times \cdots \times Y_{T_i}^{(i)}|\mathbf{x}^{(i)}, \lambda)$. For a single observed cross product $Y = Y_1 \times \cdots \times Y_T$ we get the

conditional distribution using (4.4) and (4.5)

$$p(\mathbf{y} \in Y|\mathbf{x}, \lambda) = \sum_{\mathbf{y}\in Y} p(\mathbf{y}|\mathbf{x}, \lambda) = \sum_{\mathbf{y}\in Y} \frac{p(\mathbf{y}, \mathbf{x}|\theta)}{\sum_{\mathbf{y}'\in\mathcal{Y}^T} p(\mathbf{y}', \mathbf{x}|\theta)}$$
$$= \frac{\sum_{\mathbf{y}\in Y} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right) r(\mathbf{x})}{\sum_{\mathbf{y}'\in\mathcal{Y}^T} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y'_t, y'_{t-1}, \mathbf{x})\right) r(\mathbf{x})}$$
$$= \frac{\sum_{\mathbf{y}\in Y} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right)}{\sum_{\mathbf{y}'\in\mathcal{Y}^T} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y'_t, y'_{t-1}, \mathbf{x})\right)}$$

In the same way as (4.6) we get he conditional log-likelihood with regularizer $-\sum_{k=1}^K \lambda_k^2/2\sigma^2$

(5.7)
$$L(\lambda) =$$

(5.8)
$$\sum_{n=1}^N \log \sum_{\mathbf{y}\in Y} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y_t^{(n)}, y_{t-1}^{(n)}, \mathbf{x}^{(n)})\right)$$

(5.9)
$$-\sum_{i=1}^N \log \sum_{\mathbf{y}'\in\mathcal{Y}^T} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y'_t, y'_{t-1}, \mathbf{x}^{(n)})\right)$$

(5.10)
$$-\sum_{k=1}^K \lambda_k^2/2\sigma^2$$

5.2 Gradient of Likelihood To determine the gradient of $L(\lambda)$ with respect to the unknown parameters λ_m we calculate the derivatives of the log-likelihood in equation (5.7). For notational simplicity we set $S(y_t) := \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})$. For the term in the first sum (5.8) this is

$$\frac{\partial \log\left(\sum_{\mathbf{y}\in Y} \exp\left(\sum_t \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x})\right)\right)}{\partial \lambda_m}$$
$$= \frac{\sum_{\mathbf{y}\in Y} \exp\left(\sum_t S(y_t)\right)\left(\sum_t f_m(y_t, y_{t-1}, \mathbf{x})\right)}{\sum_{\mathbf{y}\in Y} \exp\left(\sum_t S(y_t)\right)}$$
$$= \frac{\sum_{\mathbf{y}\in Y} \frac{\exp\left(\sum_t S(y_t)\right)}{\sum_{y'\in\mathcal{Y}^T} \exp\left(\sum_t S(y'_t)\right)}\left(\sum_t f_m(y_t, y_{t-1}, \mathbf{x})\right)}{\sum_{\mathbf{y}\in Y} \frac{\exp\left(\sum_t S(y_t)\right)}{\sum_{y'\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right)}}$$
$$= \frac{\sum_{\mathbf{y}\in Y} p(\mathbf{y}|\mathbf{x},\lambda)\left(\sum_t f_m(y_t, y_{t-1}, \mathbf{x})\right)}{\sum_{\mathbf{y}'\in Y} p(\mathbf{y}'|\mathbf{x},\lambda)} \quad |\text{see (4.5)}$$
$$= \sum_t \sum_{\mathbf{y}\in Y} \frac{p(\mathbf{y}|\mathbf{x},\lambda)}{\sum_{\mathbf{y}'\in Y} p(\mathbf{y}'|\mathbf{x},\lambda)} f_m(y_t, y_{t-1}, \mathbf{x})$$
$$= \sum_t \sum_{\mathbf{y}\in Y} q(\mathbf{y}|\mathbf{x},\lambda) f_m(y_t, y_{t-1}, \mathbf{x})$$

where $q(\mathbf{y}|\mathbf{x}, \lambda) := p(\mathbf{y}|\mathbf{x}, \lambda)/\sum_{\mathbf{y}'\in Y} p(\mathbf{y}'|\mathbf{x}, \lambda)$ is the relative probability of $\mathbf{y}$ with respect to Y. By using $q(\mathbf{y}|\mathbf{x}) = q(y_T, \ldots, y_{t+1}|y_t, \ldots, y_1, \mathbf{x})q(y_t, \ldots, y_1|\mathbf{x})$ and $q(y_t, \ldots, y_1|\mathbf{x}) = q(y_{t-2}, \ldots, y_1|y_t, y_{t-1}, \mathbf{x})q(y_t, y_{t-1}|\mathbf{x})$ and summing up the different parts we get

(5.11)
$$\sum_{\mathbf{y}\in Y} q(\mathbf{y}|\mathbf{x}, \lambda) f_m(y_t, y_{t-1}, \mathbf{x})$$

(5.12)
$$= \sum_{y_t,\in Y_t, y_{t-1}\in Y_{t-1}} q(y_t, y_{t-1}|\mathbf{x}, \lambda) f_m(y_t, y_{t-1}, \mathbf{x})$$

The derivative of the term in the second sum (5.9) is

$$\frac{\partial \log\left(\sum_{\mathbf{y}\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right)\right)}{\partial \lambda_m}$$
$$= \frac{\frac{\partial}{\partial\lambda_m}\left(\sum_{\mathbf{y}\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right)\right)}{\sum_{\mathbf{y}'\in\mathcal{Y}^T} \exp\left(\sum_t S(y'_t)\right)}$$
$$= \frac{\sum_{\mathbf{y}\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right) \frac{\partial}{\partial\lambda_m}\left(\sum_{t'} S(y_{t'})\right)}{\sum_{\mathbf{y}\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right)}$$
$$= \frac{\sum_{\mathbf{y}\in\mathcal{Y}^T} \exp\left(\sum_t S(y_t)\right)\left(\sum_{t'} f_m(y_{t'}, y_{t'-1}, \mathbf{x})\right)}{\sum_{\mathbf{y}'\in\mathcal{Y}^T} \exp\left(\sum_t S(y'_t)\right)}$$
$$= \sum_{\mathbf{y}\in\mathcal{Y}^T} p(\mathbf{y}|\mathbf{x},\lambda)\left(\sum_t f_m(y_t, y_{t-1}, \mathbf{x})\right)$$
$$= \sum_t \sum_{\mathbf{y}\in\mathcal{Y}^T} p(\mathbf{y}|\mathbf{x}) f_m(y_t, y_{t-1}, \mathbf{x})$$

(5.13)
$$= \sum_{y_t,\in\mathcal{Y}, y_{t-1}\in\mathcal{Y}} p(y_t, y_{t-1}|\mathbf{x},\lambda) f_m(y_t, y_{t-1}, \mathbf{x})$$

The last equation follows by using $p(\mathbf{y}|\mathbf{x}) = p(y_T, \ldots, y_{t+1}|y_t, \ldots, y_1, \mathbf{x})p(y_t, \ldots, y_1|\mathbf{x})$ and $p(y_t, \ldots, y_1|\mathbf{x}) = p(y_{t-2}, \ldots, y_1|y_t, y_{t-1}, \mathbf{x})p(y_t, y_{t-1}|\mathbf{x})$ and summing up the differerent parts.

Finally the derivative of the prior (5.10) is

(5.14)
$$\frac{\partial \sum_{k=1}^K \lambda^2/2\sigma^2}{\partial \lambda_m} = \lambda_m/\sigma^2$$

Here we assume that each feature function f_m is applicable to all pairs y_t, y_{t-1}, and does not depend on a specific t. Let $D = \{(x^{(1)}, Y^(1)), \ldots, (x^{(N)}, Y^{(N)}\}$ be the i.i.d. training set with $(x^{(n)}, Y^{(n)}) = (x^{(n)}, Y_1^{(n)} \times \cdots \times Y_T^{(n)})$.

Then the derivative of $L(\lambda)$ with respect to λ_m is

(5.15)
$$\frac{\partial L(\lambda)}{\partial \lambda_m} = \sum_{(x,Y)\in D} \Big[\sum_t \sum_{y_t,\in Y_t, y_{t-1}\in Y_{t-1}} q(y_t, y_{t-1}|\mathbf{x},\lambda) f_m(y_t, y_{t-1}, \mathbf{x}) - \sum_t \sum_{y'_t,\in \mathcal{Y}, y'_{t-1}\in \mathcal{Y}} p(y'_t, y'_{t-1}|\mathbf{x},\lambda) f_m(y'_t, y'_{t-1}, \mathbf{x}) \Big]$$

(5.16)
$$- \lambda_m/\sigma^2$$

The first term is the expected value of feature $f_m(y_t, y_{t-1}, \mathbf{x})$ under the distribution $q(\mathbf{y}|\mathbf{x},\lambda)$ of possible observations in Y. The second term, which arises from the derivative of $\log Z(\mathbf{x})$, is the expectation of f_m under the model distribution $p(\mathbf{y}|\mathbf{x},\lambda)$ with the current parameter λ. If every observation Y_t contains exactly one value y_t we have $q(y_t, y_{t-1}|\mathbf{x}) = 1$ and we are back to the usual CRF.

The calculation of $\partial L(\lambda)/\partial \lambda_m$ requires to determine the feature values $f_m(y_t, y_{t-1}, \mathbf{x})$ for every observed sequence. The probabilities are calculated for the actual parameters using the Viterbi approach [25].

5.3 Hidden Variables Recently Quattoni et al. [22] proposed a CRF with hidden variables. As for the usual CRF (4.2) they assume an input sequence $\mathbf{x} = (x_1, \ldots, x_T)$ and a sequence of unique labels $\mathbf{y} = (y_1, \ldots, y_T)$. In addition they introduce a sequence $\mathbf{h} = (h_1, \ldots, h_T)$ of hidden variables which are not observed on training examples and where each h_t is a member of a finite set $\mathcal{H}$ of possible hidden labels. For these variables they define a conditional probabilistic model

$$p(\mathbf{y}, \mathbf{h}|\mathbf{x}, \lambda) = \frac{\prod_{c\in C} \phi_c(\mathbf{y}_c, \mathbf{h}_c, \mathbf{x})}{\sum_{\mathbf{y}',\mathbf{h}'} \prod_{c\in C} \phi_c(\mathbf{y}'_c, \mathbf{h}'_c, \mathbf{x})} \tag{5.17}$$

These hidden states are able to carry relevant information over longer distances and give additional flexibility to the CRF model. The authors report a marked improvement for certain gesture recognition tasks. For the situation addressed in this paper the hidden CRF cannot be used, as it assumes a single training output and it cannot cope with a set of possible outputs. An alternative would be to model each set of possible outputs as a new state which would lead to a combinatorial explosion of states. However, along the lines used in the paper the hidden CRF may be extended to lumped outputs.

Our model can incorporate some hidden state functionality. For a given output y_t we may define a number of unobserved substates $y_{t,1}, \ldots, y_{t,r}$. During training we may may use $y_{t,1}, \ldots, y_{t,r}$ as potential outputs if output y_t is observed. The model then may decide which of the substates occurs. In this way additional long-range information may be communicated along the sequence.

6 Features Used for Supersense Tagging

The words of the whole input sequence $x = (x_1, \ldots, x_t, \ldots, x_T)$ and features derived from them 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 x_t, where a neighborhood covers the preceeding two words, the word itself, and the subsequent two words:

1. *Word*: The target word x_t itself.
2. *Lemma*: The lemma (canonical form) for the previous word x_{t-1}.
3. *Prefix*: The three-character prefix of the word x_t.
4. *Suffix*: The three-character suffix of the word x_t.
5. *First Sense*: The supersense associated to the first sense of the word x_t from WordNet.
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 an initial capital letter of x. In addition we employ the features $mcap(x_{t-1})$, $mcap(x_t)$, $mcap(x_{t+1})$ where $mcap(x)$ indicates if word x contains mixed capitalization . 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. While $top_{i,0.1}(x_t)$ indicates the possibility that topic i may be related to x_t the other features $top_{i,0.8}(x_t)$ and $top_{i,0.98}(x_t)$ indicate that topic i describes x_t well or very well respectively. This yields 150 possible features.

Table 5: Results for Noun Supersenses

Verb	Supersense Freq.		16k pot			10.6k true + 5.3k pot			16k true		
Supersense	**potential**	**true**	**Prec**	**Rec**	**F1** (StdErr)	**Prec**	**Rec**	**F1** (StdErr)	**Prec**	**Rec**	**F1** (StdErr)
act	23493	7962	81.7	80.2	80.9 (0.74)	81.2	81.2	81.2 (0.69)	81.0	80.0	80.5 (0.65)
animal	3449	1028	94.0	88.2	91.0 (0.72)	94.1	88.9	91.4 (0.62)	93.3	89.5	91.3 (0.95)
artifact	25288	8894	91.0	84.2	87.5 (0.37)	88.6	87.0	87.8 (0.26)	87.6	87.3	87.5 (0.28)
attribute	15835	4719	75.9	73.9	74.9 (0.51)	75.7	76.4	76.0 (0.73)	74.6	76.1	75.3 (0.75)
body	12369	2555	90.7	90.5	90.6 (0.42)	90.3	92.9	91.6 (0.61)	90.0	93.3	91.6 (0.51)
cognition	22302	7018	78.9	74.9	76.8 (0.57)	77.0	77.2	77.1 (0.41)	76.7	76.7	76.7 (0.61)
communication	27817	6952	80.2	82.2	81.2 (0.28)	82.3	81.8	82.1 (0.15)	82.2	81.7	81.9 (0.29)
event	7360	1806	61.5	72.7	66.6 (0.70)	68.8	68.3	68.6 (0.70)	66.9	67.0	67.0 (1.28)
feeling	2418	831	77.4	82.3	79.7 (1.08)	80.2	78.6	79.3 (1.27)	79.4	77.7	78.4 (1.42)
food	2769	600	92.3	94.3	93.3 (0.65)	91.7	92.1	91.9 (0.54)	92.4	91.1	91.7 (0.92)
group	13919	4854	78.9	84.8	81.7 (0.51)	83.8	84.6	84.2 (0.57)	85.1	84.4	84.7 (0.17)
location	10311	3558	76.4	86.7	81.2 (0.50)	79.9	85.8	82.7 (0.46)	80.8	85.1	82.9 (0.51)
motive	728	133	0.0	0.0	0.0 (0.00)	70.1	69.8	69.1 (2.63)	71.9	71.0	71.0 (1.39)
object	5671	1484	79.7	75.7	77.6 (1.00)	80.8	79.1	79.9 (0.57)	81.0	77.5	79.2 (0.70)
person	17015	8172	96.7	96.3	96.5 (0.21)	97.2	96.2	96.7 (0.22)	96.9	96.0	96.4 (0.24)
phenomenon	4256	1190	70.4	82.3	75.9 (0.86)	75.0	80.4	77.5 (1.19)	75.6	79.6	77.5 (0.86)
plant	1572	447	92.1	91.7	91.9 (1.28)	90.2	93.0	91.5 (1.25)	90.6	92.5	91.5 (1.34)
possession	4961	1483	78.5	80.0	79.3 (0.87)	84.4	81.4	82.8 (1.16)	84.5	81.9	83.1 (0.70)
process	2299	501	82.3	65.1	72.6 (1.93)	81.5	77.6	79.4 (0.93)	75.8	80.4	78.0 (1.63)
quantity	6653	1906	84.1	87.9	86.0 (0.91)	87.2	88.0	87.6 (0.49)	87.7	87.4	87.5 (0.38)
relation	3069	913	70.0	79.9	74.6 (0.74)	69.6	76.2	72.7 (0.44)	70.1	76.5	73.1 (0.90)
shape	2246	333	77.6	66.5	71.5 (2.39)	77.0	63.9	69.6 (3.75)	72.2	64.9	67.4 (3.41)
state	13968	3380	73.1	76.1	74.5 (0.78)	75.2	75.3	75.2 (0.59)	74.6	75.1	74.8 (0.41)
substance	5407	2079	83.8	91.1	87.3 (0.48)	86.6	90.3	88.4 (0.48)	87.1	89.5	88.2 (0.87)
time	7200	4158	93.7	85.8	89.6 (0.23)	91.5	92.4	91.9 (0.26)	90.5	92.6	91.5 (0.39)
Tops	12127	9785	98.1	97.3	97.7 (0.13)	98.4	97.9	98.2 (0.11)	98.3	98.0	98.2 (0.09)

The LDA model was developed by [3] 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 [2].

7 Experiments

7.1 Corpora In the SemEval Coarse Grained task [19] specific supersenses are constructed for annotation using the Oxford Dictionary of English. To be compatible with the results of [7] we instead used the original WordNet supersenses as targets and SemCor as training corpus. The Semcor dataset [18] consists of documents taken from the Brown Corpus [14]. It is freely available [24] 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 tags, lemmata and the correct WordNet synsets. In the second subset *SemcorV* of 166 documents only the verbs are annotated. We only used the first subset *SemcorN* for our experiments.

7.2 Evaluation For each experiment we have conducted a 5-fold cross validation with exactly the same partition of the SemCor Corpus ($\sim$20k sentences) into training and test data.

For each verb and noun in the data we use either the correct(annotated) supersense or a list of potential supersenses for training. These potential supersenses of a word are automatically generated by taking all possible supersenses of a word from WordNet by considering all possible senses of a word. For the noun blow, for instance, we get four potential supersenses according to table 3.

The experiments with potential labels are used to analyse the usefulness of our lumped label CRF. We analyzed a total of six different experiments:

10.6k true This experiment uses two third of the correctly

Table 6: Results for Verb Supersenses

Verb	Supersense Freq.		16k pot			10.6k true + 5.3k pot			16k true		
Supersense	**potential**	**true**	**Prec**	**Rec**	**F1** (StdErr)	**Prec**	**Rec**	**F1** (StdErr)	**Prec**	**Rec**	**F1** (StdErr)
body	8377	822	91.7	64.9	76.0 (1.37)	84.5	67.0	74.7 (1.55)	81.5	68.4	74.3 (1.63)
change	15230	4086	79.3	68.6	73.5 (0.54)	74.7	75.1	74.9 (0.81)	74.9	75.8	75.3 (0.74)
cognition	14963	4253	86.0	74.9	80.0 (0.65)	80.6	78.9	79.7 (0.70)	79.9	78.6	79.2 (0.85)
communication	18907	6336	87.1	82.1	84.5 (0.32)	85.7	83.3	84.5 (0.34)	85.9	82.7	84.3 (0.34)
competition	6164	549	62.3	62.1	62.1 (1.60)	70.9	54.9	61.7 (1.04)	64.7	58.4	61.2 (0.91)
consumption	5838	825	86.9	64.7	73.9 (4.47)	87.3	75.9	81.2 (1.18)	83.4	78.1	80.6 (1.38)
contact	13492	2693	73.7	75.4	74.5 (0.53)	73.5	76.7	75.0 (0.55)	73.0	76.1	74.5 (0.60)
creation	10344	1689	67.2	56.8	61.5 (1.40)	67.3	57.7	62.1 (1.92)	67.8	58.2	62.6 (1.56)
emotion	3960	1038	77.0	88.3	82.3 (0.51)	82.0	82.1	82.0 (0.45)	82.6	79.9	81.2 (0.49)
motion	12413	3777	71.1	86.3	77.9 (0.72)	75.0	83.1	78.8 (0.78)	74.7	82.5	78.3 (0.60)
perception	9653	2597	72.5	78.1	75.2 (0.41)	73.9	77.0	75.4 (0.42)	75.5	76.9	76.2 (0.58)
possession	19553	3031	65.5	82.5	73.0 (0.54)	68.8	78.9	73.5 (0.76)	68.2	78.7	73.0 (0.63)
social	24664	3535	61.7	74.2	67.4 (0.40)	70.9	71.7	71.3 (0.56)	71.2	70.7	70.9 (0.46)
stative	25148	12181	92.9	88.7	90.7 (0.21)	91.5	90.9	91.2 (0.24)	91.4	90.7	91.0 (0.36)
weather	428	57	40.0	6.1	10.4 (7.84)	73.2	51.3	59.5 (4.98)	76.3	50.9	61.0 (2.11)

Table 7: Overview over Results. The experiments are described below

		Macro-Average			Micro-Average		
Method	**Experiment**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
CRF	10.6k true	78.1	76.3	76.8	83.1	83.2	83.1
CRF lumped output	10.6k true + 5.3k pot	79.0	77.6	78.1	83.7	83.9	83.8
CRF	16k true	78.5	77.6	77.9	83.4	83.6	83.5
CRF lumped output	16k true & test pot	78.7	78.8	78.8	84.3	84.5	84.4
CRF lumped output	16k pot	75.6	74.4	74.4	82.8	82.7	82.8
CRF lumped output	16k pot & test pot	77.0	74.7	75.2	82.9	82.9	82.9
Ciramita & Altun (06)	16k true	-	-	-	76.6	77.7	77.1
Most frequent sense	16k true	-	-	-	63.9	69.2	66.4

labelled training data to train the CRF.

10.6k true + 5.3k pot This experiment employs two third of the correctly labelled training data and adds the other third in which we replaced the correct labels by the potential labels for each word.

16k true In the next experiment we used the complete correctly labelled training data ($\sim$16k sentences) in each fold of the cross validation.

16k pot In this experiment we used only the potential labels for the training data. This means that no manual annotations are employed during training. Therefore the training set might be increased without using human labor.

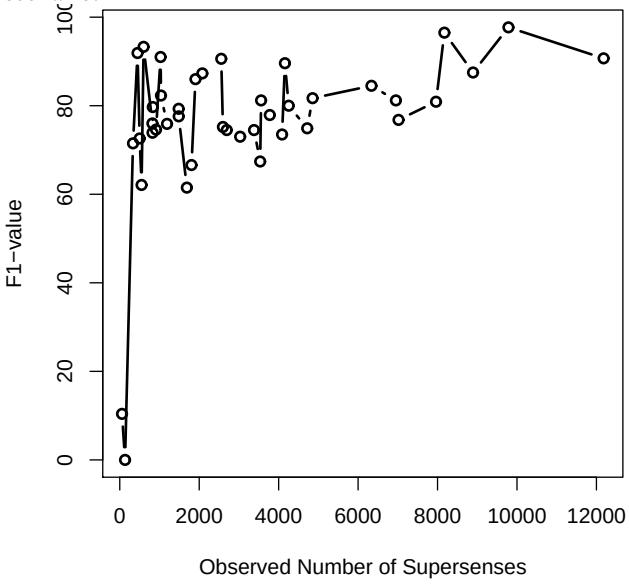
16k true & test pot In the next experiment we used the complete correctly labelled training data. In additions we automatically labelled the test data with potential labels. This may be done automatically beforehand as it involves no human annotation but only exploits the restrictions given by WordNet.

16k pot & test pot In the final experiment we used only the potential labels in the training and test set. No manual annotation at all is required for this alternative.

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 the majority of cases. A quite successful disambiguation strategy is to assign the

Figure 1: Number of Observed Supersense Labels in the SemCor Data vs. F1-value for Classification in the 16k pot scenario.



most frequent supersense to a word which can be used as a baseline.

7.3 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 [17]. For each iteration function values and gradients were calculated in parallel for different batches of data. Subsequently the final function values and gradients were computed by aggregation. In this way we could make full use of the 40 CPUs available in our cluster and reduce the computation time from about 23 days to 19 hours.

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 counts how often the supersense is a potential supersense of the word and indicates the ambiguity of the term. The remaining columns show precision, recall, F1-value and the standard error of the F1-value as determined from the variance of the crossvalidation results. The lowest F1-value is 67.3% for "motive" with the majority of values bewteen 75% and 85%. The micro-average, i.e. the average weighted by frequency of true supersense labels, is about 85%. It can be seen, that with growing frequency of a supersense there is a trend to a higher F1-value.

Table 6 contains the results for verb supersenses. Here the F1-value is somewhat lower with values between 36% and 91% with a frequency-weighted micro-average of about 80%. This shows that due to their high ambiguity supersenses of verbs are much more difficult to determine.

In table 7 the macro-averages weighted by category and the microaverages weighted by frequency are shown for different modelling approaches and experiment setups. Even if only 2/3 of the training data are used in the "10.6k true" experiment the CRF yields 83.1% micro-averaged F1, which is an increase of 6% compared to the results of Ciaramita and Altun [7]. If additional potential labels are used in the "10.6k true + 5.3k pot" case the F1-value grows to 83.8%. This increase of 6.7% corresponds to a error reduction of 29.3%.

If we compare the micro F1-value values of both experiments by a pairwise t-test, the hypothesis of equal mean values is rejected with a probability of error of 5.6%. This is a strong indication that the information gain due to potential labels increases the micro-averaged F1-value.[1] Note that the increase from 83.1% to 83.8% corresponds to a reduction of the error of about 3.6%.

If all data is used in the "16k true" experiment the F1-value no longer increases but gets insignificantly lower than in the case of potential labels. This shows that a CRF with potential labels in this case can exploit the information contained in the additional data even if the true labels are not known. We will investigate in further experiments, if the advantage of potential labels compared to the true labels may be significant.

The best result is achieved, when the true annotations are used for training and the test data is labelled with potential labels. This is possible as it involves no manual annotation but only requires the automatic derivation of the possible supersenses for each word using the WordNet structure. In this case we arrive at an F1 of 84.4% which is about 1% better than the result without introducing the restrictions.

Most interesting are the results if only potential labels are used, i.e. no manually annotated data at all. In the "16k pot" experiment we get an F1 of 82.8% which is just 1% worse than for the training data with true labels. If we also use test data with potential labels in experiment "16k pot & test pot" this does not improve the results. In this case the algorithm needs a number of observations for each supersense to infer the correct labeling. As shown in figure 1 the results for supersenses with a small number of occurences is much worse, e.g. for the supersenses "motive" and "weather" with 133 or 57 mentions respectively. As, however, no labeling is required we may easily increase the sentences with words potentially belonging to these supersenses and improve the results.

[1] In additional experiments we are currently taking into account the dependency of cross-validation samples for the significant levels according to [4].

8 Summary

We used conditional random fields to model the sequential context of words and their relation to supersenses. We extended the model so that it is able to include potential supersenses into the training data. These potential labels may be derived automatically without human annotation effort, which allows to take into account very large training sets for supersense learning.

It turns out that in our experiments the use of potential labels yields comparable results to the ones achieved by the use of the full training set with correct labels. Most interesting are the results of the experiments where the training examples of the test set with their potential labels are used in addition for training. This opens the perspective to improve the results by adding new "training" data without any labelling effort i.e. no manual annotation.

As the annotation time for a new document is quite low the approach may be used to annotate large collections of documents. As pointed out the training time may be reduced substantially by exploiting the constraints on the set of potential labels. This will be implemented in the future. In addition we will compose training sets with potential labels in such a way that all words of WordNet are represented. Finally we will investigate how to cope with new senses of words not covered by WordNet, e.g. for special domains.

9 Acknowledgement

The work presented here was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

References

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

[2] Andre Bergholz, Jeong-Ho Chang, Gerhard Paaß, Frank Reichartz, and Siehyun Strobel. Improved phishing detection using model-based features. In *Fifth Conference on Email and Anti-Spam, CEAS 2008, Aug 21-22, 2008, Mountain View, Ca*, 2008.

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

[4] Remco R. Bouckaert and Eibe Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004, Proceedings*, volume 3056 of *Lecture Notes in Computer Science*. Springer, 2004.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

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

[13] Nancy Ide and Jean Veronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1–24, 1998.

[14] H. Kucera and W. N. Francis. *Computational analysis of present-day American English*. Brown University Press, 1967.

[15] 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.

[16] D. C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

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

[18] 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.

[19] 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.

[20] Nam Nguyen and Rich Caruana. Classification with partial labels. In *KDD 2008*, pages 551–559, 2008.

[21] 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.

[22] Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10), 2007.

[23] 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.

[24] Semcor download website . http://www.cs.unt.edu/∼rada/downloads.html.

[25] 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.