Verb Sense Disambiguation using a Predicate-Argument-Clustering Model

Wiebke Wagner (wiebke.wagner@ims.uni-stuttgart.de)

Helmut Schmid (schmid@ims.uni-stuttgart.de)

Sabine Schulte im Walde (schulte@ims.uni-stuttgart.de)
Institute for Natural Language Processing, Azenbergstr. 12
D-70174 Stuttgart, Germany

Abstract

In this paper we present a verb sense disambiguation technique which is based on statistical clustering models which merge verbs with similar subcategorisation and selectional preferences into a cluster. The sense of a verb is disambiguated by (i) extracting the verb and its argument heads with a statistical parser from a given sentence, (ii) labeling the extracted verb-argument tuple with one or more clusters according to the clustering model, and (iii) assigning the verb to one of its possible senses based on this cluster information. Using only the cluster IDs as features, we obtained an accuracy of 57.06% which is close to the results of the best system in the Senseval-2 competition which used far more information. We also show that a generalization of the selectional preferences in terms of WordNet concepts leads to better performance due to a reduction of sparse data problems. **Keywords:** probabilistic verb clustering; verb sense disambiguation; selectional preferences.

Introduction

Word sense disambiguation has a long history (see (Agirre & Edmonds, 2006) for an overview) but still remains a core problem to many NLP applications such as message understanding, machine translation, and question answering. Especially the disambiguation of highly polysemous verbs with subtle meaning distinctions is difficult. The definition of sense inventories is also challenging, controversial, and not equally appropriate across NLP domains (Ide & Wilks, 2006).

High-performance Verb Sense Disambiguation (VSD) systems are trained on sense-tagged corpora and use a wide range of linguistic and non-linguistic features. The system described in (Chen & Palmer, 2009) e.g. employs a parser, a named entity tagger, and a pronoun resolver to extract syntactic features (voice, type of complements, complement heads), semantic features (WordNet synsets and hypernyms of complement heads), topical features (keywords occurring in the context), and local features (the two preceding and following words and their POS tags). A smoothed maximum entropy classifier disambiguates the sense based on these features. It achieved 64.6% accuracy on Senseval-2 data. Results on another data set (OntoNotes) with clearer sense distinctions came close to the inter-annotator agreement rate with 82.7%.

The high costs of manual semantic tagging motivated the development of semi-supervised methods. Stevenson and Joanis (2003) clustered verbs into Levin classes with an extensive feature space. Then they applied manual, semi-supervised and unsupervised approaches to automatic feature selection in order to reduce the 560 feature set to the relevant features. They reported a semi-supervised chosen set of features based on seed verbs as the most reliable choice. Lapata and Brew (2004) defined a simple probabilistic model with automatically identified verb frames, which generated preferences for Levin classes. This model was used for disambiguating polysemous verbs in terms of Levin classes. They showed that the lexical meaning of a verb determines its behaviour, particularly with respect to the choice of its arguments.

In this paper, we use a statistical clustering model which is trained on a large unlabelled corpus of verb argument tuples such as $\langle read, subj{:}obj, man, book\rangle$ which were extracted from a text corpus by means of a parser. The clusters provided by the model can be interpreted as 'sense labels'. However, these labels are unlikely to exactly match the senses of some independently defined sense inventory. Therefore the cluster labels must be mapped to these senses in order to use the clustering model for their disambiguation. The mapping is done by a statistical classifier which is trained on manually sense tagged text. The classifier computes the probability of each possible verb sense given the cluster labels.

The verb clustering model is based on the assumption that verbs which agree on their selectional preferences belong to a common semantic class. The two verbs *to sit* and *to lie* in Example 1 e.g. belong to a class of verbs which describe an entity placed on top of another entity.

(1) *The cat sits/lies on the sofa.*

Different readings of a verb usually differ in their argument preferences. Example 2 shows two readings of the verb *to roll* with different subcategorisation frames.

(2) *The thunder rolls. – Peter rolls the ton off the road.*

Example 3 demonstrates that also the class of arguments (weaponry vs. employee) can differentiate between verb meanings.

(3) *to fire a gun – to fire a manager*

These differences in subcategorisation and selectional preferences allow the clustering model to assign the readings of a verb to different clusters, which can then be used as evidence for verb sense disambiguation. We implemented a VSD system based on these ideas and evaluated it on Senseval-2 data[1].

[1] http://193.133.140.102/senseval2/, last visited June 2009

The Senseval-2 Data

The Senseval-2 shared task was a word sense disambiguation (WSD) competition for nouns, verbs and adjectives. In this paper, only the disambiguation of verbs is considered though. We tested our system on the English Lexical Sample task of the Senseval-2 data set, which contains 3565 verb instances in the training set and 1806 in the test set. This data comprises 29 different target verbs with 16.76 senses on average. This high polysemy rate is due to the fact that particle verb constructions such as *carry on* are subsumed under the base verb. Particle verbs are explicitly marked in the corpus. This facilitated disambiguation because it allowed the elimination of inappropriate readings. The Senseval-2 data are hand-tagged with one (sometimes two) WordNet sense keys of the pre-release WordNet version 1.7. The inter-tagger agreement (ITA) of the task was only 71.3% which can be taken as an upper bound for this task.

Description of the clustering models

We used two different statistical clustering models for verb-argument tuples which group the verbs based on their subcategorisation and selectional preferences. They are soft-clustering models and therefore able to assign a tuple to more than one cluster. The degree of membership in a given cluster is expressed by the conditional probability $p(cluster|tuple)$.

LSC

In the Latent Semantic Clustering (LSC) model (Rooth, Riezler, Prescher, Carrol, & Beil, 1999) the induction of clusters for a given verb-argument pair is based on the estimation of a probability distribution over tuples which consist of a cluster label, a verb and the argument heads. The LSC model is characterized by the following equation, where c is the cluster label, v is a verb and $a_1 ... a_n$ are the arguments, and $p(a|c,i)$ is the probability of the word a as the i-th argument in a tuple from cluster c.

$$p(c,v,a1,...,a_n) = p(c)p(v|c)\prod_{i=1}^{n} p(a_i|c,i) \qquad (4)$$

The cluster variable c is not observed in real data and therefore a 'hidden variable'. The LSC model assumes that the verb and the arguments are mutually independent given the cluster. In other words, it is sufficient to know that a verb belongs to some cluster c in order to predict its possible arguments. It follows that all verbs of a cluster must have similar argument preferences.

The model parameters are estimated with the EM algorithm which maximizes the likelihood of the training data consisting of verb-argument tuples without cluster information in an iterative process. After each iteration the model improves its parameters and increases the likelihood of the data. The independence assumptions mentioned above drive the clustering process because only models which approximately satisfy the independence assumptions will have a high training data likelihood. This technique is described more precisely in (Rooth et al., 1999) and (Schulte im Walde, Hying, Scheible, & Schmid, 2008). The number of clusters is predefined.

PAC

The PAC (predicate argument clustering) model (Schulte im Walde et al., 2008) is an extension of LSC. LSC considers only a fixed number of arguments from one particular subcategorisation frame, whereas PAC allows arbitrary subcategorisation frames. The tuple representation described above for LSC is augmented with a *frame argument*. The terms argument and subcategorisation frame here are used in a wider sense, since all subphrases that depend on the verb are considered as an argument phrase and belong to the subcategorisation frame; not only the obligatory ones. An example PAC tuple is given by ⟨begin, subj:obj:p:np, seller, discussion, with, buyer⟩. [2]

If f is a subcategorisation frame and n_f is the number of arguments in frame f, the PAC model is characterized by the following formula:

$$p(c,v,f,a_1,...,a_{n_f}) = p(c)p(v|c)p(f|c)\prod_{i=1}^{n_f} p(a_i|c,f,i) \qquad (5)$$

The tuple probability $p(c, read, subj{:}obj, man, book)$, for instance, is the product $p(c)p(read|c)p(subj{:}obj|c)$ $p(man|c,subj{:}obj,1)p(book|c,subj{:}obj,2)$.

Because the argument probability $p(man|c,subj{:}obj,1)$ is difficult to estimate due to sparse data problems, PAC generalizes the selectional preferences expressed in this probability distribution from words to concepts and replaces $p(man|c,subj{:}obj,1)$ by the product of a slot-specific concept probability such as $p(person|c,subj{:}obj,1)$ and a word probability such as $p(man|person)$ which is independent of the slot. The concepts are taken from a hierarchy such as WordNet. The selectional preferences of a given argument slot such as $\langle c,subj{:}obj,1\rangle$ are represented by a set of concepts which together constitute a *cut* through the WordNet hierarchy. In general, there might be more than one concept r in this set which dominates a given noun. Therefore it is necessary to sum over them:

$$p(a|c,f,i) = \sum_r p(r|c,f,i)p(a|r) \qquad (6)$$

The concept probabilities and word probabilities are not directly estimated. Instead they are derived from Markov models whose states correspond to WordNet concepts. The concept probability $p(person|c,subj{:}obj,1)$, for instance, is defined as the sum of the probabilities of all paths from *entity* to *person* in the Markov model for the slot $\langle c,subj{:}obj,1\rangle$, and the probability of a single path, in turn, is defined as the product of the state transition probabilities along that path. PAC models are trained on verb-argument tuples without cluster

[2] PP arguments contribute two elements to the frame, the preposition and the nominal head.

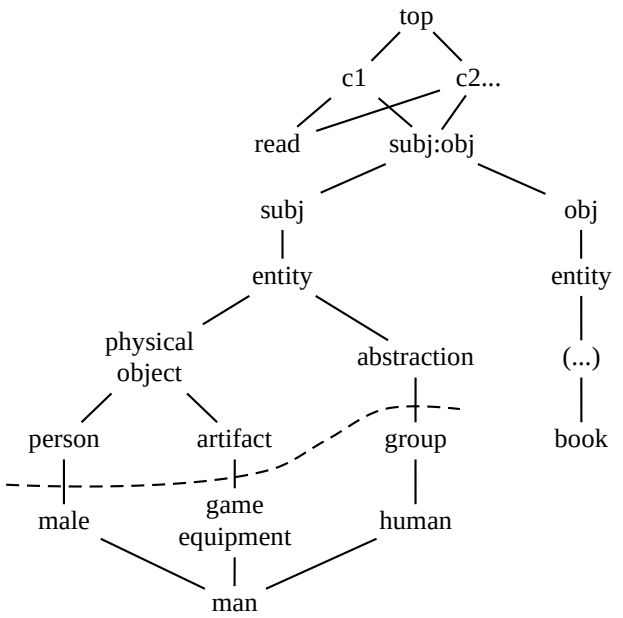

Figure 1: PAC tree of the tuple: <read, man, book>

information with a variant of the EM algorithm. Initially, the selectional preferences (SP) of the different slots only consist of the most general concept *entity*. During the training, the preferences become more specific, corresponding to a lower cut through the WordNet hierarchy. The specificity of the concepts is controlled by Minimum Description Length (MDL) pruning. In each iteration of the EM algorithm, the SP Markov models are first extended with all the hyponyms of the current terminal nodes. Then the E step and the M step of the EM training follow, and finally the resulting SP models are pruned back by eliminating all edges whose deletion decreases the total description length.

Description of the System

Our VSD system consists of two components, a clustering model (either LSC or PAC) and a classifier. It uses the verb, the subcategorisation frame, and the arguments as the only features for VSD. The clustering model is trained on the Reuters corpus[3] and learns to assign similar verbs (or actually verb readings) to the same cluster. The classifiers is trained on the Senseval-2 training corpus and learns which clusters correspond to which sense of a verb. It treats each verb separately.

Preprocessing of the Data

We parsed the Reuters corpus with the BitPar parser (Schmid, 2006) and extracted the verbs and their arguments. With the extracted tuples, we trained the verb clustering models.

The Senseval-2 corpus was also parsed with the BitPar parser but only the verbs to be disambiguated and their arguments were extracted. For each tuple, we calculated the cluster probabilities according to the verb clustering model.

$$p(c|tuple) = \frac{p(c,tuple)}{\sum_{c'} p(c',tuple)} \quad (7)$$

Cluster probabilities below a threshold of 0.1 were ignored.

Training of the Classifier

Next we used the Senseval-2 training set to train a classifier that estimates the probability of senses within a cluster. If c is a cluster and s is a sense, we first summed up the probabilities of c for any tuple that was labeled with s. This gave us the frequency of the joint occurrence of s and c.

$$f(s,c) = \sum_{tuple:sense(tuple)=s} p(c|tuple) \quad (8)$$

To get the probability of s given c, we calculated the relative frequency. The probabilities of all different senses within c therefore sum up to 1.

$$p(s|c) = \frac{f(s,c)}{\sum_s f(s,c)} \quad (9)$$

Sense Classification

The classifier assigns a sense to each tuple based on the verb, the cluster probabilities, and the sense probabilities. The most probable clusters of a tuple are obtained from the clustering model and the sense probabilities for these clusters were estimated in the training. The classifier multiplies the probability of each cluster with the probability of each sense of the cluster. The total probability of a sense for a given tuple is computed by summing over all clusters:

$$p(s|tuple) = \sum_c p(c|tuple)p(s|c) \quad (10)$$

To give an example: The cluster probabilities for the verb-argument tuple $\langle carry, subj{:}obj, man, suitcase\rangle$ might be c1=0.94, c2=0.05. The classifier would provide for c1: sense1=0.18 and sense2=0.81, whereas c2 would hold sense1=1 as a single sense. In this case, the most probable sense would be $p(s_2|tuple) = p(c_1|tuple)p(s_2|c_1) + p(c_2|tuple)p(s_2|c_2) = 0.94 * 0.81 + 0.05 * 0 = 0.76$.

In accordance with the Senseval scoring we counted each verb with an identical sense tag as a match (Kilgarriff, 2000). If no sense was found,[4] the most frequent sense (MFS) of the verb was assigned. If no MFS existed because the verb was not in the training data, we randomly chose one of the senses of the verb in WordNet1.7 and took 1 divided by the number of senses as the estimated correctness of this random decision.

[3] http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf, last visited June 2009

[4] This can be due to parsing errors or because the assigned clusters did not appear in the training data with that verb

Evaluation

The system was optimized on the training set of the English Lexical Sample task. All experiments that follow in this section are done on this data set with a tenfold cross-evaluation. We experimented with different settings of the model and of the preprocessing to find the best features.

We established a base system to explore the performance of our features. The base system uses a PAC clustering model with 50 clusters, and 100 training iterations. Additionally we compared the results to the MFS baseline which assigns all verbs to their most frequent sense.

If nouns from the verb-argument tuple were not in WordNet, we replaced them by a placeholder $\langle UNKNOWN \rangle$. Additionally we used the placeholder $\langle NONE \rangle$ when the parser failed to find the head of an argument (e.g. the subject in subject-less sentences).

For significance testing, we applied a Binomial test and considered only tuples that where classified correctly either in the base system or in the experiment system but not in both. We chose an significance threshold of 5%.

Experiments on the Data

Since the variable frame size and the conceptualisation of the arguments were an extension from LSC to PAC we aimed to discover to what extend the frames and arguments helped in the classification process. We tried to gradually increase the amount of information provided by the arguments. First we replaced the arguments in the Senseval2 and the Reuters tuples by the placeholder 'x' to use only information given from the frames. A tuple extracted from the sentence *"He began a battle"* is here represented as $\langle begin, subj{:}obj, x, x \rangle$.

In a second experiment we eliminated the generalization to concepts in PAC. This means that the probability $p(a_i|c, f, i)$ in Equation 5 is directly estimated from data and not decomposed according to Equation 6. A mapping of WordNet-unknown words is not required here. The above tuple would look as follows: $\langle begin, subj{:}obj, he, battle \rangle$

In a third experiment, we replaced pronouns that are likely to refer to humans such as *I, he, us* etc. to the WordNet concept 'person'. Other arguments which where not in WordNet were mapped to $\langle UNKNOWN \rangle$ as in the base system. Our example tuple turns into: $\langle begin, subj{:}obj, person, battle \rangle$.

Table 1 shows that the difference between no arguments at all and the base system amounts to only 2%. That means that the classification is mostly done by the subcategorisation frame. Selectional preferences improved performance just slightly. The data set where pronouns were mapped to 'person' shows the best results.

In the version without WordNet the arguments caused more damage than they helped. This was a problem of data sparseness. A given tuple with an argument *a* could only be assigned to a cluster if the model contained *a* in the same cluster, the same frame and the same slot. Because the corpus was not large enough it happened quite often that a tuple with a rare frame did not fit into any cluster. For comparison: in our 'no wordnet' data set 107 tuples out of 356 did not belong to any cluster. In the base system this happened only 21 times. This means, if we use detailed information about frames we have to generalize the nouns or we need much more data.

Table 1: Manipulating the Arguments

no arguments	53.40
no wordnet generalization	50.23
base system	55.68
pronouns to 'person'	56.88

Experiments on the Model

Number of Clusters In this experiment we trained clustering models with different numbers of clusters (see table 2)[5]. If the number of clusters was rather small, more senses were united in one cluster causing mis-classifications. An inspection of the classifier parameters of a model with 20 and 160 clusters[6] for the verb *to begin* showed that the average number of *begin*-senses in the 20 cluster model was 4.0 senses per cluster, where 13 clusters contained the verb *to begin*. The 160 cluster model had 72 clusters that contained this verb with an average number of *begin*-senses of 2.72. The total ambiguity rate of the verb *to begin* was 8.

Table 2: Variation in the Number of Clusters

c 20	54.72 (significance: 0.07)
c 40	55.90
c 50 (base system)	55.68
c 60	55.28
c 80	55.52 (significance: 0.05)
c 100	55.85
c 120	56.01
c 140	56.04
c 160	56.58 (significance: 0.07)
c 180	56.69 (significance: 0.05)
c 200	55.96

Although the results were not significant a tendency towards an improvement at higher cluster numbers was visible. It seems that the more clusters we defined the more consistent the clusters were and the better the sense classification turned out. If the number of clusters is too high, we would expect a data sparseness problem because the number of tuples per cluster decreases and the probability estimates become unreliable. Maybe this point is reached with 200 clusters.

Number of Iterations It was often observed that the performance of systems which are trained with the EM algorithm improves over a couple of iterations and then starts to decrease again. Our experiments on the number of iterations show that further training iterations did not make a significant difference after the 30th iteration (see table 3[7]). After 30

[5] Significance testing yielded values over 0.05%. Values that got close to the threshold are nominated.

[6] Only clusters with a probability over 0.01 were considered.

[7] Values marked with an asterisk are significant results compared to the base system.

iterations the results bounced up and down randomly. However, even after 100 iterations we did not reach a turning point where results got noticeably worse.

Table 3: Variation of the Number of Iterations

	c 20	c 50	c 100	c 180
i 10	51.05*	52.45*	53.38*	53.63*
i 20	54.25*	54.05*	55.06	55.06
i 30	54.50*	55.25	55.62	55.09
i 40	54.13*	55.82	55.59	56.24
i 50	54.19*	55.79	55.51	55.76
i 60	55.05*	55.68	55.42	56.01
i 70	54.38*	55.95	55.68	56.32
i 80	54.55*	55.65	55.93	56.60
i 90	54.41*	55.70	55.59	56.80*
i 100	54.72	55.68	55.85	56.69

Comparing LSC and PAC

Since the LSC model does not include the frame in its parameters and since the number of arguments must be fixed, we used a different tuple representation for LSC. We created a pseudo argument containing the frame and we chose only subject and object arguments (which are undefined if not contained in the frame): ⟨begin, subj:obj:p:np, it, visit⟩

If we applied LSC to a data set without arguments, the result was similar to the corresponding PAC result (see table 4). If we added arguments as described above, we got 50.65%. In this experiment the model was losing out because it was trained on a rather small data set[8] and had similar data sparseness problems as the PAC version without WordNet. If we used a larger training set[9], performance improved considerably (see the last row of table 4). The result shows that LSC suffers more from data sparseness than PAC which indicates that the argument generalization helps.

Table 4: Comparing LSC and PAC

	LSC	PAC
no arguments	53.07	53.40
arguments, small corpus	50.65	55.68 (base system)
arguments, large corpus	55.03	56.45

Results

The final evaluation was carried out on the test data of the English Lexical Sample task with the best combination of features according to the previous experiments. That was the data set where the pronouns were partially mapped to the WordNet concept 'person'. The model was trained on a large data set with 180 clusters and 90 iterations. Table 5 compares our results to the accuracy scores of other WSD systems on this task for verbs[10]. The performance of our system is close

Table 5: Results on the evaluation data set

MFS	46.1
Seo/Lee	57.6
Dang/Palmer	59.6
Chen/Palmer	64.6
PAC	57.06

to that of the best system in the Senseval-2 evaluation (Seo, Lee, Rim, & Lee, 2001) but somewhat behind current state of the art (Chen & Palmer, 2009). However, it must be pointed out that we used very few features - only subcategorisation frames and arguments provided from the clustering model, and that our results are likely to improve after adding further features. Seo et al. (2001)[11] used no linguistic information at all, but took into account local contexts, topical contexts and bigram contexts. These features seem to be quite different from ours. Incorporating them in our system would probably improve the performance.

Error Analysis and Future Work

We had to deal with errors on different levels. Besides of parser errors – in the Senseval-2 training set 4.1% of the target verbs were not returned – we had the problem that the information in the tuples was often incomplete. Our Senseval-2 data set contained in 2669 out of 3565 tuples one or more placeholders corresponding to arguments missing in WordNet or to unrecognised objects. If we mapped pronouns that referred to humans to the concept 'person', still 2169 tuples contained a placeholder, but results got better. This indicates that future work should concentrate on data preprocessing with anaphora resolution and named entity tagging.

To avoid the bottleneck of manually annotated training data, we would like to turn our supervised system into an unsupervised system by taking the ID of the most probable cluster as the 'verb sense'. To get an intuition of how well our system covers the senses with the clusters we chose the most frequent clusters for the verb *to begin* in a 160-cluster model and looked up the most probable senses included in these clusters. In the following, clusters and senses are listed in descending order according to the frequency or probability respectively. The verb *to begin* has eight senses in the Senseval-2 data. The MFS begin%2:30:00:: was covered in several clusters (c110, c14, c21, c26, c128), which all selected for the frame $subj$:s[12] It was interesting to see, that the clusters listed above chose different arguments. c110 selected for a location as a subject, where as c14 selected for a process, c21 for a physical object – which seems to be a very general cluster – c26 for a person and c128 for an abstraction. This means that this model fractions the sense into finer

[8] The small data set contains only tuples with words existent in WordNet (2.4 million Tuple).

[9] In the large data set all tuples provided from the Reuters corpus were taken. Words not included in WordNet were replaced by a placeholder (4.9 million tuple).

[10] Listings of the English Lexical Sample results of verbs can be found in Dang and Palmer (2002)

[11] http://www.informatics.susx.ac.uk/research/groups/nlp/mccarthy/SEVALsystems.html#kunlp, last visited June 2009

[12] 's' is a sentence slot.

grained sense distinctions than WordNet does. The sense begin%2:42:04:: was included in c119 and c75 both holding the intransitive frame and again selecting for different argument concepts: 'process' and 'person'. The sense begin%2:30:01:: is modeled about as well as the described ones.

It was more difficult to model the sense begin%2:42:00:: which occurs only 24 times out of 508 *to begin*-instances. Besides its sparseness it is very similar to sense begin%2:42:04::. The WordNet description for the former is: 'have a beginning, of a temporal event' and for the latter: 'have a beginning, in a temporal, spatial, or evaluative sense'.

Sense begin%2:42:03:: shows that our system has problems if a sense occurs with different subcategorisation frames. This sense was only tagged correctly if it occurred with the frame *subj*:*p*:*np*. It must be pointed out though that we had only 17 instances of this sense in the Senseval-2 corpus. The remaining three senses were never chosen by the system because they occurred very rarely (seven times or less).

Since selectional preferences did not improve results as much as we expected, we had a closer look at the data. Table 6 gives some examples of Senseval-2 tuples, where the first column specifies the sense, the second the subject, and the last one the object of the highly ambiguous verb *to carry*. It shows that the nouns selected by the verb, group well on a higher abstraction level. These examples indicate that selectional preferences seem to be a reasonable feature even for highly ambiguous verbs like *to carry* which encourages to improve argument extraction.

Table 6: Selectional Preferences for *to carry*

carry	subject	object
42:01	Mr. Baker (person)	weapon (artifact)
42:01	he (person)	glass (artifact)
42:02	dept (abstract)	guarantee (abstract)
42:02	bill (abstract)	ban (abstract)
42:12	woman (person)	significance (abstract)
42:12	man (person)	stigma (abstract)
42:03	plane (artifact)	bomb (instrumentality)
42:03	she (= a ship) (artifact)	rigging (instrumentality)

Summary

We proposed a verb sense disambiguation method which labels English verbs with WordNet sense keys. The system consists of (i) a clustering model which is trained on unlabelled verb-argument tuples extracted from the Reuters corpus with a parser, and (ii) a classifier which is trained on the Senseval-2 data and assigns the most likely sense to a verb. The processing consists of three steps, (i) the extraction of the target verb and its arguments with a parser, (ii) the computation of cluster probabilities for the tuple with the clustering model, and (iii) the calculation of the most probable sense based on the cluster(s) assigned in the previous step.

We used two different clustering models (LSC and PAC) and found that PAC outperformed LSC and is not quite as sensitive to data sparseness. Experiments with the number of clusters indicate that a large number of clusters tends to be better. The number of senses per cluster was found to decline as the number of clusters increases.

Our experiments on different data sets showed that information about argument heads improved results by about 2%. However, many arguments were not properly extracted or could not be mapped onto WordNet senses. The improvement resulting from the replacement of personal pronouns with the word 'person' suggests that better argument extraction methods could further increase the performance.

References

Agirre, E., & Edmonds, P. (Eds.). (2006). *Word sense disambiguation: Algorithms and applications*. Springer-Verlag. (URL: http://www.wsdbook.org/)

Chen, J., & Palmer, M. (2009). Improving english verb sense disambiguation performance with linguistically motivated features and clear sense distinction boundaries. *Language Resources and Evaluation, 34 (43/2)*, 181–208.

Dang, H., & Palmer, M. (2002). Combining contextual features for word sense disambiguation. In *Proceedings of the SIGLEX/SENSEVAL workshop on WSD: Recent success and future direction.* Philadelphia, USA.

Ide, N., & Wilks, Y. (2006). The simulation of verbal learning behavior. In E. Agirre & Ph.Edmonds (Eds.), *Word sense disambiguation: Algorithms and applications, chapter 3.* Springer.

Kilgarriff, A. (2000). Framework and results for english SENSEVAL. *Computers and Humanities, 34 (1–2)*, 15–48.

Lapata, M., & Brew, C. (2004). Verb class disambiguation using informative priors. *Computer Linguistics, 30*(2), 45–73.

Rooth, M., Riezler, S., Prescher, D., Carrol, G., & Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th Annual Meeting of the ACL.* Maryland, MD.

Schmid, H. (2006). Trace prediction and recovery with unlexicalized PCFGs and slash features. In *Proceedings of COLING-ACL'06* (pp. 177–184). Sydney, Australia.

Schulte im Walde, S., Hying, C., Scheible, C., & Schmid, H. (2008). Combining EM training and the MDL principle for an automatic verb classification incorprating selectional preferences. In *Proceedings of the 46th Annual Meeting of the ACL.* Columbus, OH.

Seo, H., Lee, S., Rim, H., & Lee, H. (2001). Kunlp system using classification information model at senseval-2. In *proceedings of the second international workshop on evaluating word sense disambiguation systems (SENSEVAL-2).* Toulouse, F.

Stevenson, S., & Joanis, E. (2003). Semi-supervised verb class discovery using noisy features. In *Proceedings of CoNLL* (pp. 71–78).