MaJo - A Toolkit for Supervised Word Sense Disambiguation and Active Learning

Ines Rehbein Josef Ruppenhofer Jonas Sunde

Universität des Saarlandes, Germany
Computational Linguistics
{rehbein,josefr}@coli.uni-sb.de s9sujona@stud.uni-saarland.de

Abstract

We present MaJo, a toolkit for supervised Word Sense Disambiguation (WSD), with an interface for Active Learning. Our toolkit combines a flexible plugin architecture which can easily be extended, with a graphical user interface which guides the user through the learning process. MaJo integrates off-the-shelf NLP tools like POS taggers, treebank-trained statistical parsers, as well as linguistic resources like WordNet and GermaNet. It enables the user to systematically explore the benefit gained from different feature types for WSD. In addition, MaJo provides an Active Learning environment, where the system presents carefully selected instances to a human oracle. The toolkit supports manual annotation of the selected instances and re-trains the system on the extended data set. MaJo also provides the means to evaluate the performance of the system against a gold standard.

We illustrate the usefulness of our system by learning the frames (word senses) for three verbs from the SALSA corpus, a version of the TiGer treebank with an additional layer of frame-semantic annotation. We show how MaJo can be used to tune the feature set for specific target words and so improve performance for these targets. We also show that syntactic features, when carefully tuned to the target word, can lead to a substantial increase in performance.

1 Introduction

An important step in Natural Language Processing is the disambiguation of word senses, without which we would not be able to interpret the meaning of an utterance or text. Word Sense Disambiguation (WSD) thus provides important information for many NLP applications in the area of information retrieval, summarisation, question answering or machine translation. To date, the best performance for the task of WSD is achieved by supervised systems which rely on manually labelled training data. However, there are two major drawbacks to the supervised approach:

1. Supervised learning requires a large amount of manually labelled data.

2. Supervised learning is highly sensitive to the domain the labelled data is taken from.

Manual annotation of large data sets is time-consuming and costly. Therefore it is infeasible to create resources a) which are large enough to capture all information needed for disambiguation and b) which are representative of all possible genres and domains we might want to work with. Active Learning is one possible approach to address this bottleneck.

Active Learning tries to reduce the amount of human annotation by carefully selecting new training instances with respect to the information content they provide for the machine learning classifier used in the supervised setting. These instances are then handed over to a human annotator, the oracle, who assigns the correct label. The basic idea is to select a small number of instances which provide important information for the classifier and to thereby achieve an increase in performance in the same range as would be achieved on a larger training set of randomly selected training examples.

Although Active Learning has been shown to be useful for WSD in general [7, 24], and in particular for tackling problems of domain adaptation [5], some open issues need to be addressed. One of them is the impact of feature design on the WSD task. Chen and Palmer [6] show that a set of rich linguistic features does improve the performance of WSD systems. Xue et al. [23] argue that word senses are partitioned along different dimensions for different verbs and that, as a consequence, we need to tune the set of features used for disambiguation for each particular target verb. What we do not know is which types of features are beneficial for which (types of) target verbs. A systematic investigation of different feature types such as syntactic and semantic features, context features, collocational features, and so on, is urgently needed.

Another issue concerns the Active Learning environment. Recent work has explored the impact of different parameters on the performance of Active Learning. Among these are the size of the seed data for initial training; different techniques for selecting new, informative examples to be labeled by the human oracle; sampling techniques for providing the system with new training instances to choose from; as well as developing stopping criteria for determining the optimal point to end the Active Learning process [24, 16, 22, 25, 2, 19]. Other issues which are also crucial for the Active Learning setup are the grain size of sense distinctions used for annotation as well as the distribution of the different word senses in the data. It is not yet clear whether Active Learning does work for coarse-grained word sense distinctions only [8], or whether it can also improve performance for fine-grained, detailed word senses [5].

In this paper, we provide a means for tackling these questions. We present MaJo, a toolkit with a graphical user interface for applying Active Learning to a Word Sense Disambiguation task. Our toolkit allows users to combine different components for syntactic and semantic pre-processing, to choose between different sets of features which can easily be adapted to the learning problem, and to integrate Active Learning in the training process. Our main intention in building Majo is to provide an explorative tool for investigating the contribution of individual features to the learning problem and to provide an easily accessible way to test the impact of Active Learning on different target words.

The remaining part of the paper is structured as follows. First we describe the architecture of our tool and show how it can be used for supervised WSD with and without Active Learning. Next we present experiments assessing the impact of syntactic and semantic features on the performace of our WSD system and show how MaJo can be used for tuning the feature set to particular target words. Finally we conclude and outline future work.

2 A Tool for Feature Exploration and Active Learning

The tool presented in the paper, MaJo, allows the user to explore the usefulness of different feature types for WSD in an Active Learning environment. The graphical user interface guides the user through the learning process and provides an easy way to include or exclude individual features for training. The ability to display the extracted features for all instances allows for a qualitative evaluation of the benefit obtained by individual features. In the Active Learning environment the user is presented with selected instances, which can comfortably be labelled in the GUI. The manually labelled instances are then added to the seed data, and the system is trained on the new data set.

2.1 Architecture

MaJo features a flexible plugin architecture which implements a number of interfaces to off-the-shelf NLP tools and linguistic resources for extracting training data from the web (Yahoo! search API), for preprocessing (Stanford POS Tagger [21], Stanford Parser [13], Berkeley Parser [20], MaltParser [18]), for extracting semantic features (WordNet [12], GermaNet [15]) and for classification (OpenNLP MaxEnt 2.5[1]). The architecture can easily be extended to incorporate additional components for preprocessing and feature extraction, or to implement new machine learning algorithms for training. At the moment the system provides working interfaces for English and German, but it can easily be extended to other languages.

2.2 Supervised Learning with MaJo

The GUI was designed to guide users with only basic computer skills through the training process. First, the user has to enter a target lemma he or she wants to train the system on. The system generates a list of possible inflected word forms for the target lemma, using a precompiled dictionary[2]. The user can remove unwanted

[1] `http://maxent.sourceforge.net`

[2] The German dictionary was created with Morphy, a morphological analyser by Wolfgang Lezius [17]

word forms or add new ones to the dictionary. Next, the user is asked to load a text file with annotated training data or to enter new, labelled instances for training in a text field. After that, the user can choose the plugins for preprocessing and feature selection.

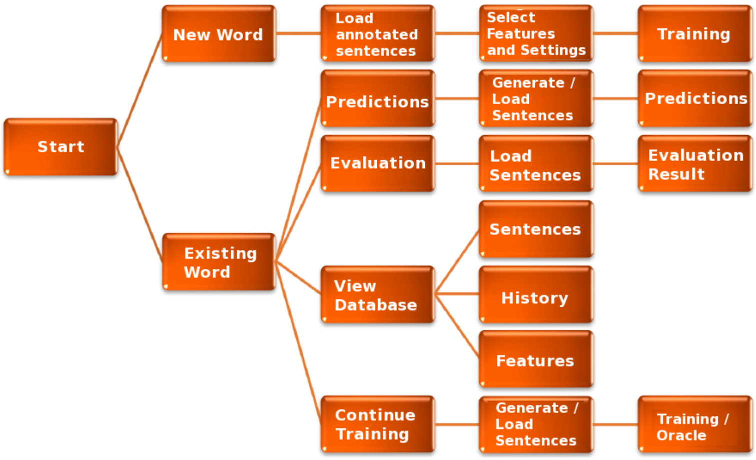

Figure 1: Working stages during supervised learning with and without Active Learning

Plugins

At the moment, our system provides the plugins listed in Table 1. Feature classes (1) and (2) are bag-of-word context features extracting the word form (1) or POS tag (2) for each word occuring in a context window of size n. Feature class (3) relies on information about syntactic categories provided by either the Berkeley Parser or the Stanford Parser, which have been trained on a constituent version of the TiGer treebank [3]. The user can specify a syntactic category, which is considered as context for which all child nodes (word form or POS tags) are extracted. Feature class (4) is based on functional dependency information provided by the MaltParser, which was trained on a dependency version of TiGer. Feature classes (5) and (6) add semantic information to the feature set, based on different preprocessing steps. For (5), we extract selected semantic relations like hyperonymy or meronymy for specified POS tags. The WordNet(/GermaNet)POSTag plugin uses a more fine-grained POS tag annotation, while the WordNet(/GermaNet)SuperordinateTag plugin uses a more coarse-grained (super-ordinate) POS tag scheme. For (6), semantic relations are extracted for specific functional dependencies as provided by the MaltParser.

After having selected the feature set and a machine learning classifier for training[3], the system starts feature extraction and training on the annotated training

[3] At present, the system implements the OpenNLP MaxEnt classifier. We plan to integrate other ML algorithms in the future.

	Feature Class	Description	Parameter	
(1)	WordRangeContext	bag-of-word context	window size	
(2)	POSTagContext	bag-of-POS-tag context	window size	Berk./Stan. POS Tagger
(3)	ClauseFunDep	words or POS tags for given functional dependencies	functional dependency	MaltParser
(4)	SentencePhrase FunDep	words or POS tags for children of a specific syntactic category	syntactic category	Berk./Stan. Parser
(5)	WordNet/GermaNet (Super)POSTag	WordNet relations for (super-ordinate) POS tags	max. depth, sem. relation	Berk./Stan. POS Tagger
(6)	WordNet/GermaNet FunDep	WordNet relations for specific functional dependencies	max. depth, sem. relation	MaltParser

Table 1: Off-the-shelf software components implemented in MaJo

data. Having finished the training process, the training data is stored in a database for future use. Now the user can access the database and evaluate the performance of the system against a gold standard (see Figure 2.2).

Accessing the database also offers other options to the user. It is possible to predict word senses for new, unannotated text which can be a) loaded from a text file, b) entered in a text field (GUI) or c) generated using the Yahoo! search interface. Another option allows the user to display the data stored in the database. For each sentence in the database the user can check which features have been extracted by the different plugins. This provides the means for a qualitative evaluation of the usefulness of individual features.

2.3 Active Learning

The last option offered by the system provides the environment for Active Learning. The pre-stored data in the database can be considered as seed data, which is used by the ML classifier to predict labels for new, unannotated text. These newly annotated sentences can serve as a pool for selecting instances for Active Learning.

As described above, for Active Learning we need to identify those instances which are the most informative for the classifier. The user can define a threshold for selecting new instances, based on the confidence score of the classifier. The confidence score reported by the maximum entropy classifier specifies the probability that instance n is assigned label x. We can use this score to determine which instances the classifier is most uncertain of. The intution behind this is that the classifier has yet to learn how to label these instances. Therefore we select new training instances by setting a confidence threshold, so that all instances below the threshold will be presented to the oracle to be manually disambiguated, and then added to the training set. The GUI provides a comfortable interface for the human annotator, who can then chose the correct label for each of the selected sentences from a pulldown menu. When the annotation is finished, the system is retrained on the new data set, consisting of the seed data and the newly added, manually labelled

freq	**bringen**	freq	**gewinnen**	freq	**drohen**
2	Position_on_a_scale				
3	Erbringen				
5	Achieve_status				
6	Deprive				
6	Put_behind				
9	Accumulated_amount				
12	Contribute_effort	3	Manufacturing		
15	Formulation	4	Improvement		
21	Entail	30	Bring_about_result		
39	Befall_patient	37	Change_position_on_a_scale		
40	Present	40	Win_prize		
105	Bringing	43	Come_to_be_in_state	243	drohen1-salsa
112	Receive_caused_experience	43	Suasion	256	Commitment
471	Cause_patient_to_be_in_state	300	Win_competition	501	Run_risk
850	train/test: 680/170	500	train/test: 400/100	1000	train/test: 800/200

Table 2: Word senses for *bringen* (to bring), *gewinnen* (to win) and *drohen* (to threaten)

instances. After retraining, the user has the option to evaluate the performance of the new training set or to continue the Active Learning process.

3 Experiments

In our experiments we want to assess the impact of different types of features on a WSD task for German verbs. We chose the three German verbs *drohen* (threaten), *gewinnen* (win) and *bringen* (bring), because they are reasonably frequent and exhibit a range of difficulty in terms of the number of word senses. Our sense inventory follows the SALSA annotation scheme [4]. The SALSA corpus is a frame-semantic lexical resource for German, adding an additional layer of semantic annotation to the TiGer treebank. Semantic frames can be considered as word senses, and so the task of frame assignment is basically a WSD problem [10].

Table 2 shows the number of instances for the three verbs in our data set as well as the different word senses and their distribution. We performed a 5-fold cross-validation, randomly generating training and test folds from the pool of available instances. The main objective of our experiments is to investigate the impact of different feature types on the WSD task for the three target verbs. We want to test how much we can gain in terms of precision and recall by tuning the feature set to the individual target verbs. Furthermore, we want to test which types of features are helpful for the different targets. What can we expect when applying shallow context features only, and how much can be gained by adding syntactic and semantic features to the feature set?

3.1 Results

In our experiments we first tested the performance of our system when trained with individual features. We report f-scores averaged over the 5 folds for each

Feature	drohen	gewinnen	bringen
A	*context features (word form)*		
wordRange 2	0.689	0.588	**0.570**
wordRange 5	**0.702**	**0.640**	0.550
wordRange 8	0.684	0.632	0.540
B	*context features (POS tags)*		
StanPOS 2	0.617	0.540	**0.500**
StanPOS 3	**0.651**	0.490	0.490
StanPOS 3, +PUNC	0.634	**0.550**	0.470
C	*Word form/POS tag context for specific syntactic categories*		
SentPhrase NP	0.491	0.550	0.430
SentPhrase VP	**0.617**	**0.642**	**0.530**
SentPhrasePOS NP	0.494	**0.598**	**0.500**
SentPhrasePOS VP	**0.621**	0.594	0.490
D	*GermaNet semantic relations for superordinate POS tags, depth 3*		
hyper, meron, N	0.570	0.612	0.530
hyper, meron, V	0.582	0.566	0.510
hyper, meron, NAV	**0.593**	**0.626**	**0.550**
E	*GermaNet semantic relations for GF (MaltParser), depth 3*		
SUBJ, OBJA, OBJD, hyper	0.499	0.612	0.510
SUBJ, OBJA, OBJD, OBJG hyper	0.510	**0.614**	0.520
SUBJ, OBJA, OBJD, hyper, meron	**0.528**	0.610	**0.550**
F	*Combinations of the best features for each target verb*		
best settings	0.701	0.650	0.560

Table 3: Results for individual feature plugins and combinations thereof

target word.

Context features

First, we trained the system using shallow context features. We extracted the word forms in a context window of size 2, 5 and 8 to the left and right of the target word. Table 3 *A* shows results for our three targets. For *drohen* and *gewinnen* we obtain best results with a context window of size 5, while for *bringen* a smaller window (size 2) shows better performance.
When extracting POS tag context features, for *drohen* best results are achieved with a context window of size 3. For *gewinnen*, including punctuation in the feature set brings a considerable improvement of 6%, while for *drohen* and *bringen* results decrease (Table 3, *B*). *Bringen*, however, achieves best results with a small window size of 2, as was the case for the word range context. Using word forms as features by far outperforms POS tag context features for all three verbs.

Syntactic features

Using syntactic categories to specify the context window for which we extract word forms, we obtain by far better results when selecting verb phrases as relevant context. For POS tag features, the verb *drohen* again benefits from the VP context (with an improved performance of around 12% over NP context), while for *gewinnen* and *bringen* we observe a slightly better performance when using NPs as context (Table 3, *C*).

Semantic features

Using semantic features as the only clue for WSD, again we obtain mixed results. Table 3, *D* shows results for extracting semantic relations from Germanet for superordinate POS tags (nouns *N*, verbs *V*, adjectives *ADJ*). While for *drohen* the contribution of the GermaNet features is less than that of the shallow context features or the syntactically motivated context features, for *gewinnen* and *bringen* the semantic features contain more relevant clues for the disambiguation process. However, they are still outperformed by the word context features which, for all three targets, obtain best results.

Our last feature class, *E*, extracts semantic relations from GermaNet for specific grammatical functions assigned by the MaltParser. Surprisingly, selecting features for functional dependents like subject, accusative object and dative object does not yield better results than extracting the same relations for all nouns, verbs and adjectives in the sentence. One possible explanation is the low performance for grammatical function labelling in German parsing. For subjects, results are quite high with 90.2% f-score, for accusative objects, however, we see only a performance of 80.0%, and for dative objects the MaltParser achieves 49.7% f-score only [14]. As a result, the classifier has to deal with a great amount of noise in the feature set, which might be responsible for the poor results for feature class *E*.

Combined features

Next, we selected the parameter setting for which we obtained best performance for each feature class, and trained the system on the combined feature set (Table 3, *F*) Surprisingly, performance for the combined features is not much higher than for the best individual feature classes. We suspect that by selecting the best performing feature setting we select features capturing the same kind of information, so that the benefit obtained from the additional features is not very great. By using a large feature set, on the other hand, we also add more noise to the data.

Dinu and Kübler [9] argue that, at least for memory-based learning, a reduced and controlled amount of features is more beneficial than the the full range of features that have been proposed in the WSD literature so far. We follow this notion and restrict our feature set to a subset of linguistically motivated features, based on an analysis of the specific sense distinctions for our target verbs. The *History* option provided by MaJo allows the user to inspect the features extracted by each of the plugins, and to use the information for feature tuning. For *drohen*, the word sense distinctions are more syntactically motivated than for *gewinnen* or *bringen*, which becomes apparent when looking at the results for the individual feature classes. The contribution of the semantic feature classes for *drohen* is rather small (as compared to results for *gewinnen* and *bringen*), while overall results for *drohen* are higher than for the other two targets.

The new set of features resulting from our analysis[4] achieves an f-score of 0.775, which is significantly higher than the result obtained by combining the sin-

[4] ClauseFunDep (Table 1, ROOT, SUBJ, OBJA, OBJD, PP, OBJP, AUX, PART, AVZ, ATTR), PosTagContext (window size 3, no punctuation, Stanford POS Tagger), WordRangeContext (window size 5), SentencePhrasePOS (Berkeley Parser, VZ)

gle best-performing feature-classes. This shows that syntactic features (as well as the semantic features based on the output of a treebank-trained dependency parser) can yield a substantial improvement when choosing the right settings for feature selection as well as appropriate features.

To compare our tool with a state-of-the-art WSD system, we trained MaJo on all instances in our *drohen* data set and tested it on a test set with 111 instances taken from the SALSA corpus. We also trained Shalmaneser [11], a shallow semantic parser, on the same training set and run it on our test set. MaJo achieves an f-score of 0.712 on the SALSA test set, while the performance of Shalmaneser is much lower with 0.362. Please note that these results are not representative for the overall performance of the two systems. For a fair comparison we need to test both systems on a larger number of target words, which is beyond the scope of this paper.

4 Discussion

The results that we have obtained match well with our linguistic intuitions about the relative difficulty of our three verbs. *Bringen* as the most polysemous verb is the most difficult, followed by *gewinnen*, and *drohen*. However, the number of senses by itself is not the whole story. Also relevant are the actual distribution of senses in the data and the ability to extract useful features.

Consider, as an example, the senses of the verb *bringen*. Several of the senses, the ones shown in Table (4), are actually multi-word expressions with easily identifiable components such as the expletive object *es*, PPs headed by specific prepositions or with reflexive pronouns as prepositional objects. However, these senses make up a mere 9% of our data. The by far most common sense, Cause_patient-_to_be_in_state, expresses a very general meaning of causation and typically occurs in the pattern *NP_acc PP bringen*, with a lot of different prepositions heading the secondary predicate expressing the caused state of affairs. The two next most frequent senses, the basic sense of Bringing and another metaphorical causation sense, Receive_caused_experience, share the same basic syntactic configuration *NP_dat NP_acc bringen* and are most readily distinguished by the semantics of the accusative object. Overall, the semantic deck is stacked against us: the frequent senses are syntactically not very distinctive, whereas the distinctive senses are not very frequent. In this regard, *bringen* contrasts rather strongly with our best-performing lemma, *drohen*. Its three senses have distinct prototypical syntactic realizations. Commitment clauses have the two forms *NP_nom drohen NP_dat PP_mit(with)* and *NP_nom drohen NP_dat VP*; drohen1-salsa has the form *NP_dat drohen NP_nom*; and Run_risk typically has the form *NP_nom drohen VP*. Importantly, the skew in the frequency of these senses is not very great, with one sense accounting for half of the tokens and the others for a quarter each.

Frame	Canonical Form	Freq.
Achieve_status	es PP_zu etwas bringen	5
Deprive	NP_acc PP_um bringen	6
Put_behind	NP_acc hinter pron_refl. bringen	21
Accumulated_amount	es PP_auf bringen	9
Formulation	NP_acc auf einen Nenner bringen	15
Entail	NP_ mit sich bringen	21
Total		77

Table 4: Easily identifiable senses of *bringen*

5 Conclusions and Future Work

We presented MaJo, a toolkit for supervised WSD, which incorporates an environment for Active Learning.[5] Our tool provides an easy-to-use GUI and is tailored to support feature tuning for specific target words such as we carried out here. Our experiments showed that, even for medium-sized data sets, much can be gained by tuning the feature set to specific target words, and that especially the syntactic and semantic feature types can bring a significant improvement, provided that we use the right features.

In future work we will extend the feature classes used for disambiguation as well as the options for the Active Learning environment, and integrate additional ML classifiers. We also plan to use the tool to study the interaction between the criteria on which sense distinctions are based and the learnability for automatic systems of these distinctions, comparing for instance FrameNet[1] sense distinctions with those of WordNet.

References

[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the COLING-ACL*, pages 86–90, 1998.

[2] Michael Bloodgood and Vijay Shanker. A method for stopping Active Learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, Boulder, Colorado, 2009.

[3] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In Erhard W. Hinrichs and Kiril Simov, editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, pages 24–42, Sozopol, Bulgaria, 2002.

[5] MaJo is freely available for research purposes and can be downloaded from `http://www.coli.uni-saarland.de/projects/salsa`.

[4] A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal. The SALSA Corpus: a German corpus resource for Lexical Semantics. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-06)*, Genoa, Italy, 2006.

[5] Yee Seng Chan and Hwee Tou Ng. Domain adaptation with Active Learning for Word Sense Disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[6] Jinying Chen and Martha Stone Palmer. Towards robust high performance Word Sense Disambiguation of English verbs using rich linguistic features. In *IJCNLP*, 2005.

[7] Jinying Chen, Andrew Schein, Lyle Ungar, and Martha Palmer. An empirical study of the behavior of Active Learning for Word Sense Disambiguation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, NY, 2006.

[8] Hoa Trang Dang. *Investigations into the role of Lexical Semantics in Word Sense Disambiguation*. PhD dissertation, University of Pennsylvania, Pennsylvania, PA, 2004.

[9] Georgiana Dinu and Sandra Kübler. Sometimes less is more: Romanian Word Sense Disambiguation revisited. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-07)*, Borovets, Bulgaria, 2007.

[10] Katrin Erk. Frame assignment as Word Sense Disambiguation. In *Proceedings of the 6th International Workshop on Computational Semantics (IWCS-6)*, Tilburg, The Netherlands, 2005.

[11] Katrin Erk and Sebastian Pado. Shalmaneser - a flexible toolbox for Semantic Role assignment. In *Proceedings of LREC*, Genoa, Italy, 2006.

[12] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (ISBN: 0-262-06197-X)*. MIT Press, first edition, 1998.

[13] Dan Klein and Chris Manning. Accurate unlexicalized parsing. In *41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan, 2003.

[14] Sandra Kübler. The PaGe 2008 shared task on parsing German. In *ACL Workshop on Parsing German (PaGe-08)*, Columbus, OH, 2008.

[15] Claudia Kunze and Lothar Lemnitzer. GermaNet - representation, visualization, application. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-02)*, 2002.

[16] Florian Laws and Hinrich Schütze. Stopping criteria for Active Learning of Named Entity Recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, August 2008.

[17] Wolfgang Lezius. Morphy - German morphology, part-of-speech tagging and applications. In *Proceedings of the 9th EURALEX International Congress*, Stuttgart, Germany, 2000.

[18] Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-06)*, Genoa, Italy, 2006.

[19] Fredrik Olsson and Katrin Tomanek. An intrinsic stopping criterion for committee-based Active Learning. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, Boulder, Colorado, 2009.

[20] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Proceedings of the Human Language Technology Conference and the 7 th Annual Meeting of the North American Chapter of the Association for Computati onal Linguistics (HLT-NAACL-07)*, Rochester, NY, 2007.

[21] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-03)*, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[22] Andreas Vlachos. A stopping criterion for Active Learning. *Compututer Speech and Language*, 22(3), 2008.

[23] Nianwen Xue, Jinying Chen, and Martha Palmer. Aligning features with sense distinction dimensions. In *ACL*, 2006.

[24] Jingbo Zhu and Eduard Hovy. Active Learning for Word Sense Disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic, 2007.

[25] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. Active Learning with sampling by uncertainty and density for Word Sense Disambiguation and Text Classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, 2008.