Semi-Supervised Learning for Word Sense Disambiguation: Quality vs. Quantity

Sandra Kübler
Indiana University
skuebler@indiana.edu

Desislava Zhekova
University of Bremen
zhekova@uni-bremen.de

Abstract

In this paper, we discuss the importance of the quality against the quantity of automatically extracted examples for word sense disambiguation (WSD). We first show that we can build a competitive WSD system with a memory-based classifier and a feature set reduced to easily and efficiently computable features. We then show that adding automatically annotated examples improves the performance of this system when the examples are carefully selected based on their quality.

Keywords

word sense disambiguation, memory-based learning, semi-supervised learning

1 Introduction

Word sense disambiguation (WSD) [7] is concerned with lexical ambiguity: It is the task of automatically assigning the correct meaning to occurrences of polysemous words in their context. Thus, word sense disambiguation is a necessary component for many applications in Computational Linguistics, such as machine translation, information retrieval, information extraction, or question answering. However, most of the WSD approaches reported so far rely on supervised learning techniques, relying on the data sets provided by the SensEval[1] and SemEval[2] competitions. This makes them susceptible for well known problems in supervised machine learning such as data sparseness or lacking domain independence.

In the present paper, we will investigate a semi-supervised approach to WSD. Semi-supervised learning starts with a supervised learner trained on available data. In a second step, data are added from automatically annotated sources. Semi-supervised approaches, especially when they do not optimize for individual words, often result in no or minimal improvements. Gonzalo and Verdejo [5] show that the good results for individual words cannot compare to supervised results. This either means that the quality of the data is not good enough to be used for the given purpose, or that the approach in employing the data is not optimal. If the former is true, future approaches to semi-supervised learning must concentrate on distinguishing reliably from unreliably annotated examples.

Our hypothesis is that the automatically annotated examples that are added to the training material are not of high enough quality to improve the results. In order to investigate the problem, we start with the SensEval-3 English lexical sample task data set[3] and then add examples extracted from a selection of lexicons and corpora. The results show that only reliably annotated examples should be considered and that they can be included in approaches different from the ones previously proposed (e.g. by Mihalcea [12]). We will show that a careful selection of automatically annotated examples gives a modest improvement over the supervised results, as compared to a significant drop in accuracy when all examples are added.

2 Related work

There is a considerable amount of work published on word sense disambiguation for English. We will concentrate here on some systems that took part in the SensEval-3 task and that we will use to place our system as well as on semi-supervised approaches.

Yarowsky [21] suggested an unsupervised self-training approach to WSD. This approach can also be used in a semi-supervised manner if the initial data is manually annotated, which results in an increase in accuracy from 90.5% for the unsupervised approach to 95.5%. A very successful approach to automatic acquisition of sense-tagged corpora is Mihalcea's [12] co-training and self-training approach. This approach is based on the creation of several (co-training) or a single (self-training) word-experts on labeled data and their further usage for labeling unannotated data. In the case that the optimal selection of parameters is chosen for each word independently, the approach for both co- and self-training strategies achieve an error reduction of 25.5%. Moreover, in order to improve the co-training method, a combination of co-training with majority voting was used. The improved co-training algorithm with a global parameter selection scheme resulted in a considerable error reduction of 9.8% as compared to the basic classifier.

Gonzalo and Verdejo [5] discuss multiple encouraging results with regard to the automatic acquisition of sense-tagged corpora. One such result is the observation that under certain circumstances, the quality of the automatically extracted data equals the quality of

[1] `http://www.senseval.org`

[2] `http://nlp.cs.swarthmore.edu/semeval/`

[3] `http://www.senseval.org/senseval3/data.html`

International Conference RANLP 2009 - Borovets, Bulgaria, pages 197–202

the manually prepared one. Another interesting remark is the fact that employing web data reaches better results than unsupervised approaches but the final system performance is still far lower than the performance of fully supervised systems.

The development of fully supervised WSD systems, however, has been given considerably more attention than the automatic acquisition of sense-tagged corpora. This is proven by approaches as the ones presented by Mihalcea et al. in the overview of the SensEval-3 English lexical sample competition [13] - naïve Bayes systems (e.g. htsa3 [6]), systems based on kernel methods (e.g. IRTS-Kernels [19]; nusels [8]), based on the Lesk algorithm [9] (e.g. wsdiit [18]), maximum entropy models (e.g. Cymfony [15]) and many others (e.g. Prob0 [17] and clr04-ls [10]). Those systems constitute the set of best performing systems from the SensEval-3 evaluation exercise.

3 The system for WSD

In order to examine evidence for our hypothesis that only automatically annotated examples of high quality can be used successfully, we designed a memory-based learning system, which we used to train multiple word-experts/classifiers, first on the SensEval training data, and in a second step with the automatically acquired data. The remainder of the section is structured as follows: In section 3.1, we will describe the supervised baseline system, section 3.2 describes the feature and parameter optimization, section 3.3 the data sets that were collected, and section 3.4 the preprocessing steps for the new data sets.

3.1 The supervised baseline system

The supervised system uses memory-based learning, in the implementation of the Tilburg Memory-Based Learner (TiMBL) [2]. Memory-based learning is a member of the k-nearest neighbor (k-NN) [14] paradigm. This approach bases the classification of a new instance on the k most similar instances found in the training data. It has been shown to be successful for a range of problems in NLP [1]. Daelemans et al. [1] argue that MBL has a suitable bias for such problems because it allows learning from atypical and low-frequency events, thus enabling a principled approach to the treatment of exceptions and sub-regularities in language. Another advantage of MBL lies in the fact that it can work with features with a large number of different values. This allows the usage of complete words as feature values. As a consequence, however, MBL is also sensitive to large numbers of features that are only relevant for the classification of specific instances but not for all instances. For this reason, the best results using memory-based learning are reached by a rather small data set, as shown by Dinu and Kübler [3] for Romanian.

The classifier is trained on the SensEval-3 English lexical sample task data set, which contains 57 ambiguous words along with examples of their use. We trained and optimized separate classifiers for individual words. The features we employed are based on the feature set by Dinu and Kübler [3]. , they are listed in Table 1. As shown in section 4.1, these features also give competitive results for English when compared to systems participating in SensEval-3.

Feature	Description
CT_{-3}	TP -3 from TW
CT_{-2}	TP -2 from TW
CT_{-1}	TP -1 from TW
CT_0	TW
CT_1	TP 1 from TW
CT_2	TP 2 from TW
CT_3	TP 3 from TW
CP_{-3}	POS of TP -3 from TW
CP_{-2}	POS of TP -2 from TW
CP_{-1}	POS of TP -1 from TW
CP_0	POS of target word
CP_1	POS of TP 1 from TW
CP_2	POS of TP 2 from TW
CP_3	POS of TP 3 from TW
NA	first noun after TW
NB	first noun before TW
VA	first verb after TW
VB	first verb before TW
PA	first preposition after TW
PB	first preposition before TW

Table 1: *Featured used for the word-experts (TP x is the token at position x and TW is the target word)*

3.2 System Optimization

Since memory-based learning is sensitive to irrelevant features (cf. e.g. [11]), it is important to optimize features for each word-expert. Following Mihalcea [11] and Dinu and Kübler [3], we used forward and backward feature selection. This resulted in a considerable reduction in the number of features actually used for individual words as well as in a considerable improvement in accuracy (see section 4.1 for details).

We also performed a non-exhaustive optimization of system parameters. The most straightforward parameters that can be optimized for a k-NN approach are the distance metric and the values for k representing the k nearest neighbors used for classifying a new instance.

We selected the overlap metric as the distance metric and tested the following values for the number of nearest neighbors, or rather the number of nearest distances in the TiMBL system: $k = 1$, $k = 3$, $k = 5$. The parameters are optimized for each word-expert individually, and only the results for the best setting are described in the results. Since the individual optimization of parameters for word-experts results in a high number of training runs, we did not explore other parameter settings, but rather used the setting found optimal for Romanian by Dinu and Kübler [3].

3.3 Data collection

For the investigation whether adding automatically annotated examples to the training set can increase coverage and consequently accuracy, we extracted examples from several dictionaries and corpora. Table 2

```
activate  [Show phonetics]
verb [T]
1 to cause something to start:
    The alarm is activated by the lightest pressure.

2 SPECIALIZED to make a chemical reaction happen more quickly, especially by heating
```

Fig. 1: *The entry for the word "activate" from the Cambridge Learner's Dictionary*

```
ascertain the size of the overdose. Evacuation of the stomach, gastric lavage, and
administration of activated charcoal. Delay in evacuating the stomach may result in
delayed absorption, leading to relapse during
```

Fig. 2: *An example for the word "activate" extracted from the BNC*

lists all dictionaries and corpora plus the number of examples extracted from these sources.While the dictionaries do contain word sense information, the sense inventories are generally different from the one used in the SensEval-3 English data set and thus cannot be used directly. For this reason, we only extracted examples without keeping track with which sense an example was associated. Figure 1 shows the entry for the word `activate` from the Cambridge Learner's Dictionary (minus the formatting). From this entry, only the example sentence for sense 1 is extracted.

Sources	No. ex.
Dictionaries	
Dictionary.com Unabridged (v 1.1)[4]	114
Cambridge Advanced Learner's Dictionary[5]	405
American Heritage Dictionary of the English Language[6]	194
The Longman Dictionary of Contemporary English Online[7]	709
WordNet 3.0[8]	300
Corpora	
British National Corpus (BNC)	15 472
ukWaC	158 072

Table 2: *The dictionaries and corpora used for the extraction of additional examples*

Additionally, we used two different corpora: the British National Corpus (BNC) and the ukWaC. The BNC contains approximately 100 million words. It was designed to be representative of current British English, both spoken and written. The second source is the ukWaC corpus [4]. The corpus was automatically constructed and consists of more than 2 billion tokens. For the examples extracted from the corpora, no sense information was available. We extracted a context of maximally 100 words. In most cases, the actual text was shorter, restricted by the corpus interface. Figure 2 shows an example from the BNC for the word `activate`.

Annotation	Source	Train	Tests
Manual	Senseval-3	7 860	3 944
	Dictionaries	1 722	
Automatic	BNC	15 472	
	ukWaC	158 072	
Total		183 126	3 944

Table 3: *The collection of examples in our system*

Table 3 gives an overview of the data sources used in the experiments, the set amounts to approximately 183 000 instances for training. For testing, we used the original test set from the SensEval-3 competition to make our results comparable to previous work.

3.4 Data preprocessing

For the additional examples, we had to preprocess the text to change the representation so that it was comparable to the SensEval examples. For preprocessing, punctuation was stripped off, words were tokenized, and meta characters were deleted. Additionally, we used a POS tagger to assign morpho-syntactic annotation to the words. For this purpose, we used the POS tagger by Tsuruoka and Tsujii [20], which is accurate and very efficient, a definite concern with regard to the text size of the newly acquired examples.

In a next step, the examples from the dictionaries and corpora had to be annotated for senses. For this purpose we used the generalized framework for WSD - `WordNet::SenseRelate::TargetWord`[9] developed by Pedersen, Patwardhan, and Banerjee [16]. `WordNet::SenseRelate::TargetWord` uses a modification of the Lesk algorithm [9] that also includes glosses of related words from WordNet. The best sense for a word is then selected based on its semantic relatedness to these words from the context and from WordNet. Based on a manual evaluation of a small data set, we chose the local module for determining relatedness.

The reason for not using self-training, as Mihalcea [12] did, was that we wanted to avoid a bias towards

[4] http://dictionary.reference.com

[5] http://dictionary.cambridge.org

[6] http://education.yahoo.com/reference/dictionary

[7] http://www.ldoceonline.com

[8] http://wordnetweb.princeton.edu/perl/webwn

[9] http://www.d.umn.edu/ tpederse/senserelate.html

majority senses in the training data. We intend the additional examples to complement the available training set from SensEval-3. This means, we are interested in adding examples for minority senses where at all possible. If we used our main machine learning approach, to which we will add these examples in the end, it would show a tendency to annotate the new examples with majority senses. For this reason, we chose a method that is based on semantic similarity measures.

4 Experiments

We investigate the question whether automatically annotated examples can help a supervised system. Our hypothesis is that such additional examples can be beneficial if we can select high quality examples from the large pool with high confidence. For this reason, we present five sets of experiments: 1) The supervised experiment in which only the designated SensEval-3 training set is used. 2) A lower bound experiment in which we use the complete set of automatically annotated examples as the training set. This will show that the sheer size of the automatically annotated examples is not sufficient for gaining competitive results. 3) The first semi-supervised approach, in which all automatically examples are added to the training set. 4) The second semi-supervised experiment, in which the set of automatically annotated examples is sampled based on the sense distribution of the SensEval-3 training set. 5) The third semi-supervised experiment, the set of automatically annotated examples is sampled based on quality.

The results of the system were evaluated by the SensEval-3 scoring software *scorer2*[10]. The scorer calculates precision and recall, for both fine-grained and coarse-grained evaluations. However, since our classifier assigns senses to all instances, we will present only accuracy, which is equivalent to precision and recall.

	Experiment	Coarse	Fine
	MFS [13]	64.5	55.2
1)	supervised	79.3	75.1
2)	unsupervised	56.2	47.5
3)	semi-supervised: all	64.5	57.8
4)	semi-supervised: ratio	77.5	72.7
5)	semi-supervised: quality	79.4	75.0

Table 4: *The results of the experiments*

4.1 The Supervised WSD System

The results of the system (using the features presented in section 3.1) when trained only on the SensEval-3 training set are shown in row 1 in Table 4.

A comparison to the best participating systems in the third SensEval evaluation exercise, shown in Table 5, confirms that our system is highly competitive. Note that our results are based on a much more restricted feature set than the ones used by all the other system. For example, we do not use any context features apart from the closest nouns, verbs, and prepositions. Instead, the features are all easily and efficiently computable. It is noteworthy that our system with the restricted feature set reaches a fine-grained accuracy that is significantly higher than that of all participating systems. These results give us a good basis for using our system in further experiments with data that has been automatically gathered and annotated.

System/Team	Coarse	Fine
nusels/Nat.U.Singapore	78.8	72.4
htsa3/U.Bucharest	79.3	72.9
IRST-Kernels/ITC-IRST	**79.5**	72.6
our system	79.3	**75.1**

Table 5: *Comparison with the three best supervised systems in the Senseval-3 lexical sample task [13]*

4.2 Training on automatically annotated examples

The experiment described here uses all examples collected from the dictionaries and from the corpora. As described in Section 3.4, they were automatically annotated by `WordNet::SenseRelate::TargetWord`. The whole set of examples was then used as training data for the memory-based classifier, using the same feature set as for the baseline system. The results of this experiment are shown in row 2 of Table 4.

The results show that the accuracy based on this training set is considerably lower than that for the supervised approach with the same feature set. As expected, a larger size of training data, without quality control, is not sufficient for reaching good results in WSD.

4.3 Semi-supervised WSD with all automatically annotated examples

In this experiment, we used the SensEval-3 training set and added all the automatically annotated examples. The results of this experiment are shown in row 3 of Table 4. The surprising result here is that adding more examples does not improve accuracy. On the contrary, the results are approximately 15 percent points lower than the baseline results in the coarse evaluation and approximately 17 percent points for the fine-grained evaluation. These results corroborate earlier findings that adding more examples is not always beneficial.

4.4 Semi-supervised WSD with filtering based on ratio

One reason for the disappointing results in the previous section might be that the sense distribution of the newly added examples does not correspond to the distribution in the SensEval-3 data. For this reason, we conducted one experiment in which we filtered the automatically annotated data so that the sense distribution in the SensEval-3 training set is maintained. We

[10] `http://www.cse.unt.edu/~rada/senseval/senseval3/scoring/scoring.tar.gz`

are aware that the sense distribution in the training set does not necessarily correspond to the distribution in the test set. But since in a real-world setting, there is no possibility of determining the sense distribution in the test set short of manually annotating it, we use the distribution in the training set as the best estimate available to us.

In the present experiment, filtering is defined as excluding all examples that violate the distribution of senses in the SensEval-3 training set. To clarify the procedure, we will use a simple example. Let us consider one of the words in the training set - the verb `suspend`. Let us suppose that in the original manually annotated training set, this word appears with 3 senses and the following distribution: 19, 12, and 48. We record this distribution and filter our automatically prepared examples in such a way that the senses for the word have the same proportional distribution. If we assume that we have 39, 60, and 100 additional examples for the three senses, then the maximal ratio for sense 1 is 2, which restricts us to adding 38, 24, and 96 examples.

The results of adding the ratio filtered examples to the original training set are shown in row 4 of Table 4. Filtering the training set in order to preserve the distribution of senses as it is in the SensEval-3 lexical sample task training set improves results considerably when we compare this experiment to the one using all automatically annotated examples: Accuracy improves by 13 percent points in the coarse evaluation and by 15 percent points in the fine-grained evaluation. This comparison shows that the new examples have a radically different sense distribution than the SensEval data. It also shows that obtaining a similar sense distribution to the one in the test data is of utmost importance. However, the results are still below the results for supervised learning, which shows that the sense distribution is not the only factor that needs to be taken into account when adding new examples.

4.5 Semi-supervised WSD with filtering based on quality

Since filtering based on ratio only partially helps closing the gap between the supervised system and the one with all examples, we maintain our hypothesis that the quality of the added examples must be high in order for them to be useful in classifying the test data. To test this hypothesis, we carried out a final experiment, in which we filtered the new examples based on their quality.

Filtering based on quality works as follows: First, we have the supervised baseline system classify each of the instances from our automatically annotated example set. We determine the quality of this example by its distance to the nearest neighbors in the original SensEval training set. The distance is provided by TiMBL. Second, based on the distances, we extract only those instances that differ only minimally (distance < 2) from the manually annotated training set. We then add the resulting collection of examples (consisting of 141 instances) to the SensEval training set. The experiment achieves the results shown in row 5 of Table 4.

The results for this experiment show a slight improvement in the coarse evaluation over the supervised baseline. However, we have to keep in mind here that we only added a minimal number of examples (141). This means that the number of examples added for an individual word never exceeds 6 instances so that the overall changes in accuracy are only minimal. The reason for this is that we concentrated on using only the examples of the highest quality. In order to show the differences that adding these few examples makes, we show the result for those words for which new examples were added in Table 6.

The results show that for seven words, the results improve for both types of evaluation. In three cases, only the coarse evaluation improves, and in two cases, the fine-grained evaluation. The highest improvement is reached for the word `add`, for which adding 2 examples results in an improvement of both scores from 84.8% to 87.5%. Apart from the improvements, however, we also have a decrease in performance for seven words.

5 Conclusion and future work

We described the design and performance of a memory-based word sense disambiguation system with a very limited feature set, which makes use of automatic feature selection and minimal parameter optimization. We showed that the system performs competitively to other state-of-art systems, and we used it further for the evaluation of automatically acquired data for word sense disambiguation.

We also investigated the extension of the supervised training set by extracting additional examples from online lexicons and corpora, which are then annotated automatically with a WSD system based on semantic distance. Adding these examples, however, results in a dramatic drop in performance. Filtering the additional examples to maintain the original distribution of senses improves results, but they still do not reach the quality of supervised training only. However, filtering the additional examples for quality does increase overall performance slightly. This corroborates our hypothesis that additional examples can only be used successfully if they are of high quality. Since this method still results in a decrease in performance for several words, we are planning to refine the definition of how to determine the quality of examples in the future.

References

[1] W. Daelemans, A. van den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–43, 1999. Special Issue on Natural Language Learning.

[2] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. TiMBL: Tilburg memory based learner – version 6.1 – reference guide. Technical Report ILK 07-07, Induction of Linguistic Knowledge, Tilburg University, 2007.

[3] G. Dinu and S. Kübler. Sometimes less is more: Romanian word sense disambiguation revisited. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2007*, Borovets, Bulgaria, 2007.

[4] A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 6th Language Resources and*

Word	Baseline	Filtered	Baseline		Filtered	
	no. ex.	no. ex.	coarse	fine	coarse	fine
activate	228	234	83.8	83.8	84.2	84.2
add	263	265	84.8	84.8	87.5	87.5
appear	265	271	76.7	76.7	76.7	76.7
arm	266	272	90.2	90.2	90.2	90.2
ask	261	262	64.1	64.1	63.4	63.4
atmosphere	161	165	71.0	71.0	69.8	69.8
audience	200	203	97.0	79.5	99.0	79.5
bank	262	267	88.6	80.7	87.9	79.9
decide	122	128	85.5	85.5	82.3	82.3
degree	256	261	86.7	77.0	89.8	78.9
difference	226	230	66.2	60.1	70.2	58.8
eat	181	187	90.8	90.8	92.0	92.0
encounter	130	136	96.9	72.3	96.9	70.8
expect	156	162	87.2	87.2	86.5	86.5
express	110	111	89.1	80.0	89.1	80.0
interest	185	191	75.3	73.7	71.0	71.0
note	132	138	79.1	79.1	79.1	79.1
organization	112	118	87.5	81.2	89.3	80.4
party	230	234	72.8	71.6	72.8	72.0
plan	166	171	88.7	88.7	88.1	86.9
produce	186	192	68.1	67.0	69.1	69.1
provide	136	142	98.6	95.7	100.0	97.1
remain	139	145	92.9	92.9	94.3	94.3
shelter	196	202	72.4	72.4	68.4	68.4
sort	190	196	87.0	75.5	87.0	75.5
talk	146	151	79.5	79.5	79.5	79.5
watch	100	102	92.2	80.4	92.2	82.4

Table 6: *System results for individual words under the supervised and the semi-supervised condition with quality-based filtering*

Evaluation Conference (LREC 2008), Marrakech, Morocco, May 2008.

[5] J. Gonzalo and F. Verdejo. Automatic Acquisition of Lexical Information and Examples. In E. Agirre and P. Edmonds, editors, *Word Sense Disambiguation*, pages 253–274. Springer, 2007.

[6] C. Grozea. Finding optimal parameter settings for hight performance word sense disambiguation. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 125–128, Barcelona, Spain, July 2004.

[7] N. Ide and J. Véronis. Word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.

[8] Y. K. Lee, H. T. Ng, and T. K. Chia. Supervised word sense disambiguation with Support Vector Machines and multiple knowledge sources. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, July 2004.

[9] M. Lesk. Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, Toronto, Canada, 1986.

[10] K. C. Litkowski. SENSEVAL-3 task automatic labeling of semantic roles. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12, Barcelona, Spain, 2004.

[11] R. Mihalcea. Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics, COLING'02*, Taipeh, Taiwan, 2002.

[12] R. Mihalcea. Co-training and self-training for word sense disambiguation. In *Proceedings of the Conference on Natural Language Learning (CoNLL 2004)*, Boston, MA, 2004.

[13] R. Mihalcea, T. Chklovski, and A. Kilgarriff. The SENSEVAL-3 English lexical sample task. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain, 2004.

[14] H. T. Ng and H. B. Lee. Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 40–47, Santa Cruz, CA, 1996.

[15] C. Niu, W. Li, R. K. Srihari, H. Li, and L. Crist. Context clustering for word sense disambiguation based on modeling pairwise context similarities. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 187–190, Barcelona, Spain, July 2004.

[16] S. Patwardhan, S. Banerjee, and T. Pedersen. SenseRelate::TargetWord - a generalized framework for word sense disambiguation. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 1692–1693, Pittsburgh, Pennsylvania, USA, 2005.

[17] J. Preiss. Probabilistic WSD in SENSEVAL-3. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 213–216, Barcelona, Spain, 2004.

[18] G. Ramakrishnan, B. Prithviraj, and P. Bhattacharyya. A gloss-centered algorithm for disambiguation. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 217–221, Barcelona, Spain, 2004.

[19] C. Strapparava, A. Gliozzo, and C. Giuliano. Pattern abstraction and term similarity for word sense disambiguation: IRST at senseval-3. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 2004.

[20] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of HLT/EMNLP*, pages 467–474, Vancouver, Canada, 2005.

[21] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA, 1995.