OntoNotes: Corpus Cleanup of Mistaken Agreement Using Word Sense Disambiguation

Liang-Chih Yu and **Chung-Hsien Wu**
Dept. of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.
{lcyu,chwu}@csie.ncku.edu.tw

Eduard Hovy
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292, USA
hovy@isi.edu

Abstract

Annotated corpora are only useful if their annotations are consistent. Most large-scale annotation efforts take special measures to reconcile inter-annotator disagreement. To date, however, no-one has investigated how to automatically determine exemplars in which the annotators agree but are wrong. In this paper, we use OntoNotes, a large-scale corpus of semantic annotations, including word senses, predicate-argument structure, ontology linking, and coreference. To determine the mistaken agreements in word sense annotation, we employ word sense disambiguation (WSD) to select a set of suspicious candidates for human evaluation. Experiments are conducted from three aspects (*precision*, *cost-effectiveness ratio*, and *entropy*) to examine the performance of WSD. The experimental results show that WSD is most effective on identifying erroneous annotations for highly-ambiguous words, while a baseline is better for other cases. The two methods can be combined to improve the cleanup process. This procedure allows us to find approximately 2% remaining erroneous agreements in the OntoNotes corpus. A similar procedure can be easily defined to check other annotated corpora.

1 Introduction

Word sense annotated corpora are useful resources for many natural language applications. Various machine learning algorithms can then be trained on these corpora to improve the applications' effectiveness. Lately, many such corpora have been developed in different languages, including SemCor (Miller et al., 1993), LDC-DSO (Ng and Lee, 1996), Hinoki (Kasahara et al., 2004), and the sense annotated corpora with the help of Web users (Chklovski and Mihalcea, 2002). The SENSEVAL[1] (Kilgarriff and Palmer, 2000; Kilgarriff, 2001; Mihalcea and Edmonds, 2004) and SemEval-2007[2] evaluations have also created large amounts of sense tagged data for word sense disambiguation (WSD) competitions.

The OntoNotes (Pradhan et al., 2007a; Hovy et al., 2006) project has created a multilingual corpus of large-scale semantic annotations, including word senses, predicate-argument structure, ontology linking, and coreference[3]. In word sense creation, sense creators generate sense definitions by grouping fine-grained sense distinctions obtained from WordNet and dictionaries into more coarse-grained senses. There are two reasons for this grouping instead of using WordNet senses directly. First, people have trouble distinguishing many of the WordNet-level distinctions in real text, and make inconsistent choices; thus the use of coarse-grained senses can improve inter-annotator agreement (ITA) (Palmer et al., 2004; 2006). Second, improved ITA enables machines to more accurately learn to perform sense tagging automatically. Sense grouping in OntoNotes has been calibrated to ensure that ITA averages at least 90%. Table 1 shows the OntoNotes sense

[1] http://www.senseval.org
[2] http://nlp.cs.swarthmore.edu/semeval
[3] Year 1 of the OntoNotes corpus has been released by Linguistic Data Consortium (LDC) (http://www.ldc.upenn.edu) in early 2007. The Year 2 corpus will be released in early 2008.

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1057–1064
Manchester, August 2008

Sense Tag	Sense Definition	WordNet sense
arm.01	The forelimb of an animal	WN.1
arm.02	A weapon	WN.2
arm.03	A subdivision or branch of an organization	WN.3
arm.04	A projection, a narrow extension of a structure	WN.4 WN.5

Table 1. OntoNotes sense tags and definitions. The WordNet version is 2.1.

Example sentence:	
The 45-year-old Mr. Kuehn, who has a background in crisis **management**, succeeds Alan D. Rubendall, 45.	
management.01:	*Overseeing or directing. Refers to the act of managing something.* He was given overall management of the program. I'm a specialist in risk management. The economy crashed because of poor management.
management.02:	*The people in charge. The ones actually doing the managing.* Management wants to start downsizing. John was promoted to Management. I spoke to their management, and they're ready to make a deal.

Table 2. Example sentence for the target word *management* along with its sense definitions.

tags and definitions for the word *arm* (noun sense). The OntoNotes sense tags have been used for many applications, including the SemEval-2007 evaluation (Pradhan et al., 2007b), sense merging (Snow et al., 2007), sense pool verification (Yu et al., 2007), and class imbalance problems (Zhu and Hovy, 2007).

In creating OntoNotes, each word sense annotation involves two annotators and an adjudicator. First, all sentences containing the target word along with its sense distinctions are presented independently to two annotators for sense annotation. If the two annotators agree on the same sense for the target word in a given sentence, then their selection is stored in the corpus. Otherwise, this sentence is double-checked by the adjudicator for the final decision. The major problem of the above annotation scheme is that only the instances where the two annotators disagreed are double-checked, while those showing agreement are stored directly without any adjudication. Therefore, if the annotators happen to agree but are both wrong, then the corpus becomes polluted by the erroneous annotations. Table 2 shows an actual occurrence of an erroneous instance (sentence) for the target word *management*. In this example sentence, the actual sense of the target word is *management.01*, but both of our annotators made a decision of *management.02*. (Note that there is no difficulty in making this decision; the joint error might have occurred due to annotator fatigue, habituation after a long sequence of *management.02* decisions, etc.)

Although most annotations in OntoNotes are correct, there is still a small (but unknown) fraction of erroneous annotations in the corpus. Therefore, some cleanup procedure is necessary to produce a high-quality corpus. However, it is impractical for human experts to evaluate the whole corpus for cleanup. Given that we are focusing on word senses, this study proposes the use of WSD to facilitate the corpus cleanup process. WSD has shown promising accuracy in recent SENSEVAL and SemEval-2007 evaluations.

The rest of this work is organized as follows. Section 2 describes the corpus cleanup procedure. Section 3 presents the features for WSD. Section 4 summarizes the experimental results. Conclusions are drawn in Section 5.

2 Corpus Cleanup Procedure

Figure 1 shows the cleanup procedure (dashed lines) for the OntoNotes corpus. As mentioned earlier, each word along with its sentence instances is annotated by two annotators. The anno-

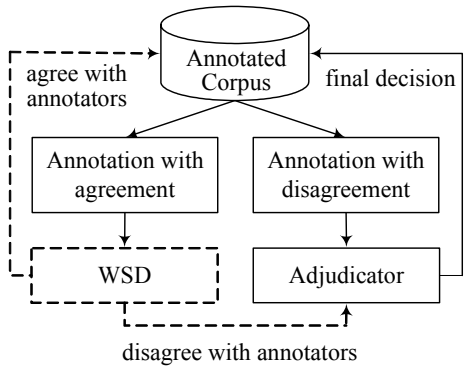

Figure 1. Corpus cleanup procedure.

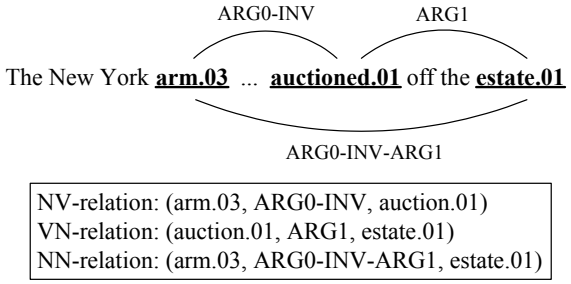

Figure 2. Example of predicate-argument structure. The label "-INV" denotes an inverse direction (i.e., from a noun to a verb).

tated corpus can thus be divided into two parts according to the annotation results. The first part includes the annotation with disagreement among the two annotators, which is then double-checked by the adjudicator. The final decisions made by the adjudicator are stored into the corpus. Since this part is double-checked by the adjudicator, it will not be evaluated by the cleanup procedure.

The second part of the corpus is the focus of the cleanup procedure. The WSD system evaluates each instance in the second part. If the output of the WSD system disagrees with the two annotators, the instance is considered to be a suspicious candidate, otherwise it is considered to be clean and stored into the corpus. The set of suspicious candidates is collected and subsequently evaluated by the adjudicator to identify erroneous annotations.

3 Word Sense Disambiguation

This study takes a supervised learning approach to build a WSD system from the OntoNotes corpus. The feature set used herein is similar to several state-of-the-art WSD systems (Lee and Ng., 2002; Ando, 2006; Tratz et al., 2007; Cai et al., 2007; Agirre and Lopez de Lacalle, 2007; Specia et al., 2007), which is further integrated into a Naïve Bayes classifier (Lee and Ng., 2002; Mihalcea, 2007). In addition, a new feature, predicate-argument structure, provided by the OntoNotes corpus is also integrated. The feature set includes:

Part-of-Speech (POS) tags: This feature includes the POS tags in the positions (P_{-3}, P_{-2}, P_{-1}, P_0, P_1, P_2, P_3), relative to the POS tag of the target word.

Local Collocations: This feature includes single words and multi-word n-grams. The single words include (W_{-3}, W_{-2}, W_{-1}, W_0, W_1, W_2, W_3), relative to the target word W_0. Similarly, the multi-word n-grams include ($W_{-2,-1}$, $W_{-1,1}$, $W_{1,2}$, $W_{-3,-2,-1}$, $W_{-2,-1,1}$, $W_{-1,1,2}$, $W_{1,2,3}$).

Bag-of-Words: This feature can be considered as a global feature, consisting of 5 words prior to and after the target word, without regard to position.

Predicate-Argument Structure: The predicate-argument structure captures the semantic relations between the predicates and their arguments within a sentence, as shown in Figure 2. These relations can be either *direct* or *indirect*. A direct relation is used to model a verb-noun (VN) or noun-verb (NV) relation, whereas an indirect relation is used to model a noun-noun (NN) relation. Additionally, an NN-relation can be built from the combination of an NV-relation and VN-relation. For instance, in Figure 2, the NN-relation (R3) can be built by combining the NV-relation (R1) the VN-relation (R2). Therefore, the two features, R1 and R3, can be used to disambiguate the noun *arm*[4].

4 Experimental Results

4.1 Experiment setup

The experiment data used herein was the 35 nouns from the SemEval-2007 English Lexical Sample Task (Pradhan et al., 2007b). All sentences containing the 35 nouns were selected from the OntoNotes corpus, resulting in a set of 16,329 sentences. This data set was randomly split into training and test sets using different proportions (1:9 to 9:1, 10% increments). The WSD systems (described in Section 3) were then

[4] Our WSD system does not include the sense identifier (except for the target word) for word-level training and testing.

	Baseline (MFS)	WSD								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy	0.696	0.751	0.798	0.809	0.819	0.822	0.824	0.831	0.836	0.832

Table 3. Accuracy of the baseline and WSD systems with different training portions.

	Baseline (MFS)	WSD								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Prec	0.090 (17/188)	0.113 (20/177)	0.112 (16/143)	0.113 (17/150)	0.124 (16/129)	0.123 (15/122)	0.127 (16/126)	**0.142** (17/120)	0.130 (14/108)	0.125 (14/112)

Table 4. Cleanup precision of the baseline and WSD systems with different training portions.

built from the different portions of the training set, called WSD_1 to WSD_9, respectively, and applied to their corresponding test sets. In each test set, the instances with disagreement among the annotators were excluded, since they have already been double-checked by the adjudicator. A baseline system was also implemented using the principle of *most frequent sense* (MFS), where each word sense distribution was retrieved from the OntoNotes corpus. Table 3 shows the accuracy of the baseline and WSD systems.

The output of WSD may agree or disagree with the annotators. The instances with disagreement were selected from each WSD system as suspicious candidates. This experiment randomly selected at most 20 suspicious instances for each noun to form a suspicious set of 687 instances. An adjudicator who is a linguistic expert then evaluated the suspicious set, and agreed in 42 instances with the WSD systems, indicating about 6% (42/687) truly erroneous annotations. This corresponds to 2.6% (42/16329) erroneous annotations in the corpus as a whole, which we verified by an independent random spot check.

In the following sections, we examine the performance of WSD from three aspects: precision, cost-effectiveness ratio, and entropy, and finally summarize a general cleanup procedure for other sense annotated corpora.

4.2 Cleanup precision analysis

The cleanup precision for a single WSD system can be defined as the number of erroneous instances identified by the WSD system, divided by the number of suspicious candidates selected by the WSD system. An erroneous instance refers to an instance where the annotators agree with each other but disagree with the adjudicator. Table 4 lists the cleanup precision of the baseline and WSD systems. The experimental results show that WSD_7 (trained on 70% training data) identified 17 erroneous instances, out of 120 selected suspicious candidates, thus yielding the highest precision of 0.142. Another observation is that the upper bound of WSD_7 was 0.35 (42/120) under the assumption that it identified all erroneous instances. This low precision discourages the use of WSD to automatically correct erroneous annotations.

4.3 Cleanup cost-effectiveness analysis

The cleanup procedure used herein is a semi-automatic process; that is, WSD is applied in the first stage to select suspicious candidates for human evaluation in the later stage. Obviously, we would like to minimize the number of candidates the adjudicator has to examine. Thus we define a metric, the cost-effectiveness (CE) ratio, to measure the performance of WSD. The *cost* rate is defined as the number of suspicious instances selected by a single WSD system, divided by the total number of suspicious instances in the suspicious set. The *effectiveness* rate is defined as the number of erroneous instances identified by a single WSD system, divided by the total number of erroneous instances in the suspicious set. In this experiment, the baseline value of the cost-effectiveness ratio is 1, which means that human expert needs to evaluate all 687 instances in the suspicious set to identify 42 erroneous instances. Figure 3 illustrates the CE ratio of the WSD systems. The most cost-effective WSD system was WSD_7. The CE ratios of the baseline and WSD_7 are listed in Table 5. The experimental results indicate that 17.5% of suspicious instances were required to be evaluated to identify about 40% erroneous annotations when using WSD_7.

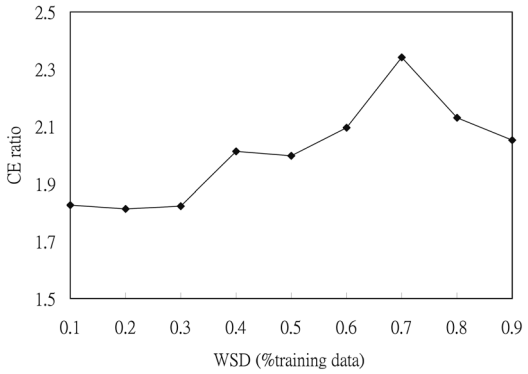

Figure 3. CE ratio of WSD systems with different training portions.

	Cost	Effect	CE Ratio
Baseline (MFS)	0.274 (188/687)	0.405 (17/42)	1.48
WSD_7	0.175 (120/687)	0.405 (17/42)	2.31

Table 5. CE ratio of the baseline and WSD_7.

4.4 Entropy analysis

So far, the experimental results show that the best WSD system can help human experts identify about 40% erroneous annotations, but it still missed the other 60%. To improve performance, we conducted experiments to analyze the effect of word entropy with respect to WSD performance on identifying erroneous annotations.

For the SemEval 35 nouns used in this experiment, some words are very ambiguous and some words are not. This property of ambiguity may affect the performance of WSD systems on identifying erroneous annotation. To this end, this experiment used *entropy* to measure the ambiguity of words (Melamed, 1997). The entropy of word can be computed by the word sense distribution, defined as

$$H(W) = -\sum_{ws_i \in W} P(ws_i)\log_2 P(ws_i), \qquad (1)$$

where $H(W)$ denotes the entropy of a word W, and ws_i denotes a word sense. A high entropy value indicates a high ambiguity level. For instance, the noun *defense* has 7 senses (see Table 7) in the OntoNotes corpus, occurring with the distribution {.14, .18, .19, .08, .04, .28, .09}, thus yielding a relative high entropy value (2.599). Conversely, the entropy of the noun *rate* is low (0.388), since it has only two senses with very skewed distribution {.92, .08}.

Consider the two groups of the SemEval nouns: the nouns for which at least one (Group 1) or none (Group 2) of their erroneous instances can be identified by the machine. The average entropy of these two groups of nouns was computed, as shown in Table 6. An independent *t-test* was then used to determine whether or not the difference of the average entropy among these two groups was statistically significant. The experimental results show that WSD_7 was more effective on identifying erroneous annotations occurring in highly-ambiguous words ($p<0.05$), while the baseline system has no such tendency ($p=0.368$).

Table 7 shows the detail analysis of WSD performance on different words. As indicated, WSD_7 identified the erroneous instances (7/7) occurring in the two top-ranked highly-ambiguous nouns, i.e., *defense* and *position*, but missed all those (0/12) occurring in the two most unambiguous words, i.e., *move* and *rate*. The major reason is that the sense distribution of unambiguous words is often skew, thus WSD systems built from such imbalanced data tend to suffer from the over-fitting problem; that is, tend to over-fit the predominant sense class and ignore small sense classes (Zhu and Hovy, 2007). Fortunately, the over-fitting problem can be greatly reduced when the entropy of words exceeds a certain threshold (e.g., the dashed line in Table 7), since the word sense has become more evenly distributed.

4.5 Combination of WSD and MFS

Another observation from Table 7 is that WSD_7 identified more erroneous instances when the word entropy exceeded the cut-point, since the over-fitting problem was reduced. Conversely, MFS identified more ones when the word entropy is below the cut-point. This finding encourages the use of a combination of WSD_7 and MFS for corpus cleanup; that is, different strategies can be used with different entropy intervals. For this experiment data, MFS and WSD_7 can be applied below and above the cut-point, respectively, to select the suspicious instances for human evaluation. As illustrated in Figure 4, when the entropy of words increased, the accumulated effectiveness rates of both WSD_7 and MFS increased accordingly, since more erroneous instances were identified. Additionally, the difference of the accumulated effect rate of MFS

	Group 1	Group 2	Difference	p-value
Baseline (MFS)	1.226	1.040	0.186	0.368
WSD_7	1.401	0.932	0.469*	0.013

*p<0.05

Table 6. Average entropy of two groups of nouns for the baseline and WSD_7.

Noun	#sense	Major Sense	Entropy	#err. instances	WSD_7	MFS	WSD_7+ MFS
defense	7	0.28	2.599	5	5	4	5
position	7	0.30	2.264	2	2	2	2
base	6	0.35	2.023	1	1	0	1
system	6	0.54	1.525	2	1	0	1
chance	4	0.49	1.361	1	1	1	1
order	8	0.72	1.348	4	1	0	1
part	5	0.70	1.288	1	1	1	1
power	3	0.51	1.233	3	1	3	3
area	3	0.72	1.008	2	1	2	2
management	2	0.62	0.959	2	1	0	0
condition	3	0.71	0.906	1	0	1	1
job	3	0.78	0.888	1	0	0	0
state	4	0.83	0.822	1	0	0	0
hour	4	0.85	0.652	1	1	1	1
value	3	0.90	0.571	2	1	1	1
plant	3	0.88	0.556	1	0	0	0
move	4	0.93	0.447	6	0	0	0
rate	2	0.92	0.388	6	0	1	1
Total	—	—	—	42	17	17	21

Nouns without erroneous instances: *authority, bill, capital, carrier, development, drug, effect, exchange, future, network, people, point, policy, president, share, source, space*

Table 7. Entropy of words versus WSD performance. The dashed line denotes a cut-point for the combination of the baseline and WSD_7.

and WSD_7 increased gradually from the beginning until the cut-point, since MFS identified more erroneous instances than WSD_7 did in this stage. When the entropy exceeded the cut-point, WSD_7 was more effective and thus its effectiveness rate kept increasing, while that of MFS increased slowly, thus their difference was decreased with the rise of the entropy. For the combination of MFS and WSD_7, its effectiveness rate before the cut-point was the same as that of MFS, since MFS was used in this stage to select the suspicious set. When WSD was used after the cut-point, the effectiveness rate of the combination system increased continuously, and finally reached 0.5 (21/42).

Based on the above experimental results, the most cost-effective way for corpus cleanup is to use the combination method and begin with the most ambiguous words, since the WSD system in the combination method is more effective on identifying erroneous instances occurring in highly-ambiguous words and these words are also more important for many applications. Figure 5 shows the curve of the CE ratios of the combination method by starting with the most ambiguous word. The results indicate that the CE ratio of the combination method decreased gradually after more words with lower entropy were involved in the cleanup procedure. Additionally, the CE ratio of the combination method was improved by using MFS after the cut-point and finally reached 2.50, indicating that 50% (21/42) erroneous instances can be identified by double-checking 20% (137/687) of the suspicious set. This CE ratio was better than 2.31 and 1.48, reached by WSD_7 and MFS respectively.

The proposed cleanup procedure can be applied to other sense annotated corpora by the following steps:

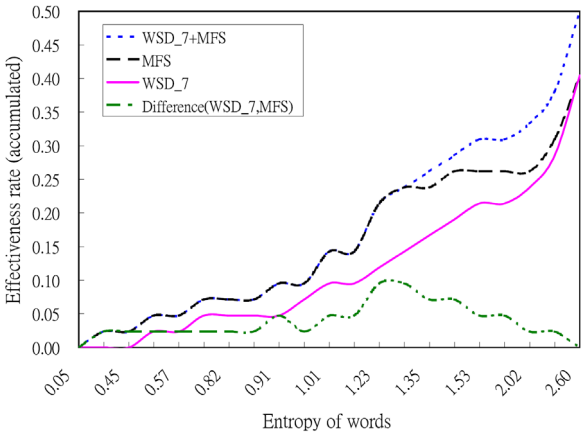

Figure 4. Effectiveness rate against word entropy.

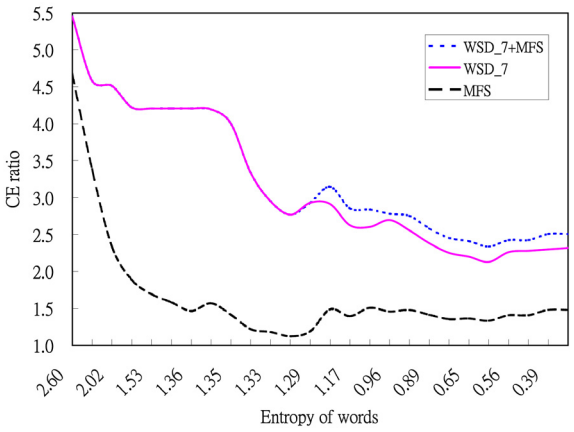

Figure 5. CE ratio against word entropy.

- Build the baseline (MFS) and WSD systems from the corpus.
- Create a suspicious set from the WSD systems.
- Calculate the entropy for each word in terms of it sense distribution in the corpus.
- Choose a cut-point value. Select a small portion of words with entropy within a certain interval (e.g., 1.0 ~ 1.5 in Table 7) for human evaluation to decide an appropriate cut-point value. The cut-point value should not be too low or too high, since WSD systems may suffer from the over-fitting problem if it is too low, and the performance would be dominated by the baseline system if it is too high.
- Combine the baseline and best single WSD system through the cut-point.
- Start the cleanup procedure in the descending order of word entropy until the CE ratio is below a predefined threshold.

5 Conclusion

This study has presented a cleanup procedure to identify incorrect sense annotation in a corpus. The cleanup procedure incorporates WSD systems to select a set of suspicious instances for human evaluation. The experiments are conducted from three aspects: precision, cost-effectiveness ratio, and entropy, to examine the performance of WSD. The experimental results show that the WSD systems are more effective on highly-ambiguous words. Additionally, the most cost-effective cleanup strategy is to use the combination method and begin with the most ambiguous words. The incorrect sense annotations found in this study can be used for SemEval-2007 to improve the accuracy of WSD evaluation.

The absence of related work on (semi-) automatically determining cases of erroneous agreement among annotators in a corpus is rather surprising. Variants of the method described here, replacing WSD for whatever procedure is appropriate for the phenomenon annotated in the corpus (sentiment recognition for a sentiment corpus, etc.), are easy to implement and may produce useful results for corpora in current use. Future work will focus on devising an algorithm to perform the cleanup procedure iteratively on the whole corpus.

References

E. Agirre and O. Lopez de Lacalle. 2007. UBC-ALM: Combining k-NN with SVD for WSD. In *Proc. of the 4th International Workshop on Semantic Evaluations (SemEval-2007) at ACL-07*, pages 342-345.

R.K. Ando. 2006. Applying Alternating Structure Optimization to Word Sense Disambiguation. In *Proc. of CoNLL*, pages 77-84.

J.F. Cai, W.S. Lee, and Y.W. Teh. 2007. Improving Word Sense Disambiguation Using Topic Features. In *Proc. of EMNLP-CoNLL*, pages 1015-1023.

T. Chklovski and R. Mihalcea. 2002. Building a Sense Tagged Corpus with Open Mind Word Expert. In *Proc. of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions at ACL-02*, pages 116-122.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

E.H. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. OntoNotes: The 90% Solution. In *Proc. of HLT/NAACL-06*, pages 57-60.

K. Kasahara, H. Sato, F. Bond, T. Tanaka, S. Fujita, T. Kanasugi, and S. Amano. 2004. Construction of a

apanese Semantic Lexicon: Lexeed. In *IPSG SIG: 2004-NLC-159*, Tokyo, pages 75-82.

A. Kilgarriff. 2001. English Lexical Sample Task Description. In *Proc. of the SENSEVAL-2 Workshop*, pages 17-20.

A. Kilgarriff and M. Palmer, editors. 2000. SENSEVAL: Evaluating Word Sense Disambiguation Programs, *Computer and the Humanities*, 34(1-2):1-13.

Y.K. Lee and H.T. Ng. 2002. An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In *Proc. of EMNLP*, pages 41-48.

I.D. Melamed. 1997. Measuring Semantic Entropy. In *Proc. of ACL-SIGLEX Workshop*, pages 41-46.

R. Mihalcea. 2007. Using Wikipedia for Automatic-Word Sense Disambiguation. In *Proc. of NAACL/HLT-07*, pages 196-203.

R. Mihalcea and P. Edmonds, editors. 2004. In *Proc. of SENSEVAL-3*.

G. Miller, C. Leacock, R. Tengi, and R. Bunker. 1993. A Semantic Concordance. In *Proc. of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308.

H.T. Ng and H.B. Lee. 1996. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach. In *Proc. of the 34th Meeting of the Association for Computational Linguistics (ACL-96)*, pages 40-47.

M. Palmer, O. Babko-Malaya, and H.T. Dang. 2004. Different Sense Granularities for Different Applications. In *Proc. of the 2nd International Workshop on Scalable Natural Language Understanding at HLT/NAACL-04*.

M. Palmer, H.T. Dang, and C. Fellbaum. 2006. Making Fine-grained and Coarse-grained Sense Distinctions, Both Manually and Automatically. *Journal of Natural Language Engineering*, 13:137–163.

S. Pradhan, E.H. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2007a. OntoNotes: A Unified Relational Semantic Representation. In *Proc. of the First IEEE International Conference on Semantic Computing (ICSC-07)*, pages 517-524.

S. Pradhan, E. Loper, D. Dligach, and M. Palmer. 2007b. SemEval-2007 Task 17: English Lexical Sample, SRL and All Words. In *Proc. of the 4th International Workshop on Semantic Evaluations (SemEval-2007) at ACL-07*, pages 87-92.

R. Snow, S. Prakash, D. Jurafsky, and A.Y. Ng. 2007. Learning to Merge Word Senses. In *Proc. of EMNLP-CoNLL*, pages 1005-1014.

L. Specia, M. Stevenson, and M. das Gracas V. Nunes. 2007. Learning Expressive Models for Word Sense Disambiguation. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, pages 41–48.

S. Tratz, A. Sanfilippo, M. Gregory, A. Chappell, C. Posse, and P. Whitney. 2007. PNNL: A Supervised Maximum Entropy Approach to Word Sense Disambiguation. In *Proc. of the 4th International Workshop on Semantic Evaluations (SemEval-2007) at ACL-07*, pages 264-267.

L.C. Yu, C.H. Wu, A. Philpot, and E.H. Hovy. 2007. OntoNotes: Sense Pool Verification Using Google N-gram and Statistical Tests. In *Proc. of the OntoLex Workshop at the 6th International Semantic Web Conference (ISWC 2007)*.

J. Zhu and E.H. Hovy. 2007. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem, In *Proc. of EMNLP-CoNLL*, pages 783-790.