Crowdsourcing WordNet

Chris Biemann
Powerset, a Microsoft company
475 Brannan St 330
San Francisco, CA 94107, USA
cbiemann@microsoft.com

Valerie Nygaard
Powerset, a Microsoft company
475 Brannan St 330
San Francisco, CA 94107, USA
vnygaard@microsoft.com

Abstract

This paper describes an experiment in using Amazon Mechanical Turk to collaboratively create a sense inventory. In a bootstrapping process with massive collaborative input, substitutions for target words in context are elicited and clustered by sense; then more contexts are collected. Contexts that cannot be assigned to a current target word's sense inventory re-enter the loop and get a supply of substitutions. This process provides a sense inventory with its granularity determined by common substitutions rather than by psychologically motivated concepts. Evaluation shows that the process is robust against noise from the crowd, yields a less fine-grained inventory than WordNet and provides a rich body of high precision substitution data at a low cost.

1 Introduction

Disappointing progress in Word Sense Disambiguation (WSD) competitions has often been attributed to problems with WordNet (Miller et al., 1990). While a valuable resource to the NLP community, WordNet was not originally designed for WSD. Still being the best option available, it became the standard resource in Senseval and Semeval competitions. High WSD performance scores using WordNet suffer from the extremly fine-grained distinctions that characterize the resource and by the relatively little available data for senses in contexts (cf. e.g. (Agirre and Edmonds, 2006)) . For example, of the eight noun senses of "hook", four refer to a *bent, curvy object*. However, in the entire SemCor (Mihalcea, 1998) there is only one occurrence recorded for this sense altogether, so for most of the senses the only data available are the glosses and the relations to other synsets. Even if some fine-grained classes are combined by clustering WordNet senses (Mihalcea and Moldovan, 2001), alignment of the sense inventory and the target domain or application remains a problem. Using WordNet, or any predefined inventory, for WSD may result in a mismatch with the target domain of the application. If it does not fit well, domain adaptation will be required, a costly endeavor that will likely have to be repeated. Corpus-based word sense acquisition, on the other hand, guarantees a match between inventory and target domain.

A major potential application of WSD is to supply correct substitutions in context for ambiguous words. The ability to make the right substitutions, in turn, gives rise to fuzzy semantic matching in Information Retrieval. However, as (Sanderson, 1994) estimated, at least 90% accuracy is required before the benefits of WSD-supplied substitutions or term expansions outweigh the drawbacks from errors.

Since semantic annotation tasks are notoriously difficult and low inter-annotator agreement goes hand in hand with low WSD scores, the OntoNotes project (Hovy et al., 2006) aimed at high agreement through word sense grouping to mitigate the problems above. In (Hovy et al., 2006) it was shown that enforcing more than 90% inter-annotator agreement and more coarse-grained sense groupings in fact can ensure accuracy levels close to 90%. However, manual methods of sense reduction such as those used in OntoNotes are costly and may not be scalable due to their dependence on highly trained annotators working for extended periods to learn the annotation protocols.

In this work, we pursue another path: we set up a bootstrapping process that relies on redundant

human input to crystallize a word sense inventory for given target words and a target corpus. Instructions are kept short and tasks are kept simple. Because some degree of noisy input is tolerated in this method, naiive annotators can be used, and data can be produced quickly, at low cost and without access to a full in-house annotation staff.
As a crowdsourcing platform, we use Amazon Mechanical Turk (AMT). AMT allows requesters to post arbitrary tasks, which are done for pay by a large pool of annotators. AMT allows to specify the number of annotators per task item, as well as to restrict the annotator pool by various criteria. The quality of annotations from AMT has been shown to be comparable to a professional annotators when answers from four or more different annotators are combined, see (Snow et al., 2008).

The remainder of this paper is organized as follows. First, we describe the three different crowdsourcing tasks in detail. Then, we lay out an overall system that connects these steps in a bootstrapping cycle carefully motivating our design decisions.
Finally, we lay out an experiment we conducted with this system, provide a quantitative and a qualitative analysis and evaluation and describe the resource resulting from this experiment.

2 Three Turker Tasks

This section is devoted to three elementary tasks given to annotators whom we will refer to as *turkers* in the AMT platform. The nature of crowdsourcing makes it necessary to follow some guidelines when designing tasks: (1) Both tasks and instruction sets for those tasks must be simple to hold training to a minimum, (2) redundancy is necessary to assure quality. The inherent noisiness of the process requires that only answers supplied multiple times by different turkers should be accepted. Requiring redundancy in answers is also important in identifying deliberate scammers.

2.1 Task 1: Finding Substitutions

The rationale behind this task is to be able to identify word senses by the differences in possible substitutions. For information retrieval applications, we find this substitution-based definition of senses desirable. Here, WSD is input for determining which lexical expansions should be used for matching, so the concept of substitutability is central.
In this task, turkers are presented with a sentence containing a target word emphasized in bold. They are asked to supply possible substitutions for the bolded word in the specific sentential context. Turkers must supply at least one, and up to a maximum of five substitutions. This task is very similar to the task used in (McCarthy and Navigli, 2007). In addition, turkers are asked to state whether the sentence is a good or acceptable representative of the target word meaning. When turkers indicate that assigning a sense to the target is hard or impossible, part-of-speech violations and insufficient contexts are potential culprits.

2.2 Task 2: Aligning Senses

This task measures how similar the senses of two usages of the same word are. The rationale behind this task is to measure closeness of senses and to be able to merge senses in cases where they are identical. Turkers are shown a pair of sentences that contain the same emphasized target word. They are then asked whether the meaning of this target word in the two sentences is identical, similar, different or impossible to determine.

2.3 Task 3: Match the Meaning

This task presents a sentence with a bolded target word and requires turkers to align usages of the target word to a given inventory: they are asked to choose one of a set of sentences containing the target word in different meanings representing the current sense inventory. They also can state that the sense is not covered by the current inventory, or label the sense as impossible to determine. To make the task more efficient and easier, this assignment contains ten of these questions for the same word using the same inventory, so the annotator has to understand the inventory choice only once.

3 Bootstrapping a Word Sense Inventory

This section describes how a word sense inventory is constructed using the tasks in Section 2. Figure 1 provides a schematic overview of how the three tasks are executed in sequence. Note that each target word is processed separately, and that the process is described for a single target word. We will use the noun target ”station” for exemplifying the

steps.
When using untrained turkers, the process that distills their raw responses to usable data must be formulated in a way that uses redundant answers and is robust against noise. Various thresholds described below have been introduced for these reasons. When trained, professional annotators are used, most of these constraints could be relaxed.

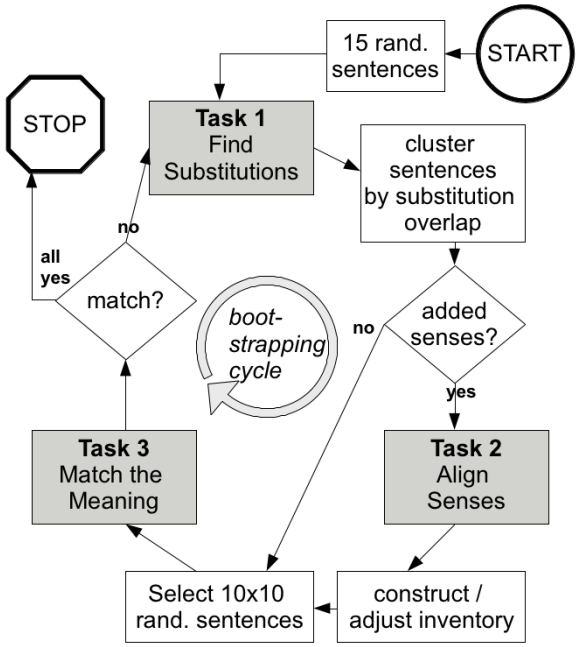


Figure 1: Bootstrapping process with Turker tasks

3.1 From Start to Task 1

For the target word, 15 random sentences are selected from a corpus resource. While 15 sentences is a somewhat arbitrary number, we have found that it ensures that the major senses of the target word are usually covered. These sentences are seeds to initiate the cycle, and need not cover all the senses in the corpus or final inventory.
In our experiments, we select by lemma and part-of-speech, so e.g. for the noun target "station" we would select sentences where "station" or "stations" were tagged as a noun. These sentences are presented in Task 1. We will use these three sentences as an example:

- **A:** The train left the *station*.
- **B:** This radio *station* broadcasts news at five.
- **C:** Five miles from the *station*, the railway tracks end.

These were assigned the following substitutions (multiplicity in brackets):

- **A:** terminal(5), railway station(3), rail facility(1), stop(1)
- **B:** radio station(4), network(3), channel(2)
- **C:** terminal(3), stop(2), train depot(2), railway station(1)

3.2 From Task 1 to Task 2

Having obtained a set of substitutions from the crowd, we compute a weighted similarity graph with the sentences as nodes. Edge weights are given by the amount of overlap in the substitutions of sentences. If two sentences share at least 3 different keywords, then their similarity is given by the sum of the common keywords they share. In the example, sentences A and C would get a score of 8 (terminal) + 4 (railway station) + 3 (stop) = 15. We only use sentences that are good or acceptable representatives for this task (as judged by turkers).
Using Chinese Whispers (Biemann, 2006), we apply graph clustering on this graph to group sentences that have a high similarity assuming that they contain the target word in the same meaning. We have chosen Chinese Whispers because it has been shown to be useful in sense clustering before (by e.g. (Klapaftis and Manandhar, 2008)) and has the property of finding the number of clusters automatically. This is crucial since we do not know the number of senses a priori. Note, however, that we do not use the outcome of the clustering directly, but ask turkers to validate it as part of the next step in the cycle, as described in the next section.
We exclude clusters consisting of singleton sentences. For each cluster, we select as the most prototypical representative sentence, the sentence that has the highest edge weight sum within the cluster. Ties are broken by evaluating difficulty and length (shorter is better). This sentence plus the substitutions of the cluster serves as sense inventory entry. In case this steps adds no senses to the inventory or the clustering resulted in only one sense, we continue with Task 3, otherwise, we validate the inventory.

3.3 From Task 2 to Task 3

The danger of clustering is either that two senses are merged or one sense is split into two entries. While merging merely results in more bootstrapping cycles, to ensure that the lost meaning not

represented by the prototypical sentence is recovered, avoiding multiple entries per sense is more serious and must be dealt with directly. This is why we present all possible pairs of prototypical sentences in Task 2. If the majority of turkers judge that two sentences have identical meanings, we merge the entries, choosing the representative sentence at random.
Then we retrieve 100 random sentences containing our target word, group them in sets of ten and present them in Task 3.

3.4 Closing the loop

All sentences that could be matched to the inventory by the majority of turkers are added to the resource. Sentences for which the sense of the target word could not be determined due to disagreement are set aside. Sentences that are marked as uncovered by the current inventory re-enter the loop and we retrieve substitutions for them in Task 1. In our example, these might be sentences like

- **D:** The mid-level *station* is situated at 12400ft altitude.
- **E:** They were desperately looking for a gas *station*.

Those sentences will probably display a high overlap in substitutions with other sentences of the current or previous iterations. In this way, additional senses are identified and verified against the current inventory.
Only if almost none (we use a threshold of three) of the 100 sentences are marked as uncovered, we estimate that the vast majority of the senses in the corpus are represented in our inventory, and the process terminates for the target word.

4 Experiment

4.1 Experimental Setup

The choice of the underlying corpus was determined by our target application, a semantic search on Wikipedia. We used a sentence-broken, POS-tagged version of English Wikipedia (dump from January 3rd, 2008) and applied a few filters to ensure complete sentences of reasonable length. Due to its diversity in topics, Wikipedia works well as a corpus for word sense acquisition. We ran the bootstrapping acquisition cycle on the 397 most frequent nouns, implementing the crowdsourcing part with Amazon Turk. During the annotation process of the first 50 words, the data supplied by five turkers per task was heavily curated and scammers were identified and blocked manually.
The most productive and reliable ten turkers were then invited to perform the annotation of the remaining 347 words with virtually no curation or manual intervention. With these trusted turkers, a redundancy of three assignments per task was shown to be sufficient. With the current set of trusted turkers, we were able to reach a speed of 8 words per day for the overall process. The speed can be increased by simply adding more annotators.

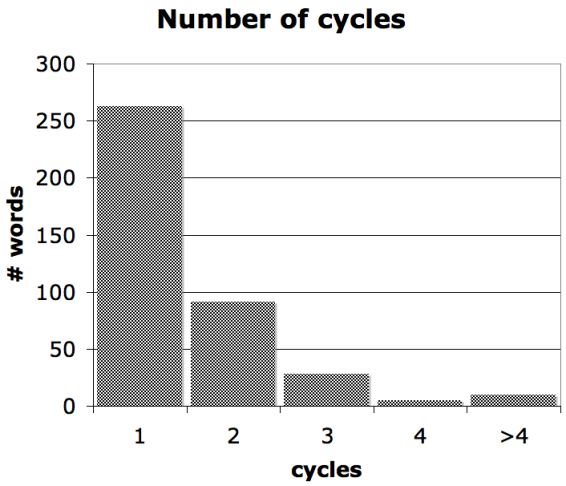


Figure 2: Distribution: words per number of cycles until convergence. On average, a word needed 1.56 cycles to converge.

4.2 Quantitative Analysis

Figure 2 shows the distribution of the number of cycles the words needed to converge. About two thirds of all words need only one cycle to complete, only ten words entered the cycle more than four times. The run for two words was finally terminated after ten iterations (see Section 5.3)..
Taking a look at the granularity of the inventory, Figure 3 shows the distribution of the number of words per number of senses. Even when using the highest frequencies nouns, almost half of the words are assigned only one sense, and over 90% of words have fewer than five senses.

For learning approaches to word sense disambiguation it is important to know how much data is available per sense. Figure 4 provides the distribution of sentences per sense in the resource. Minor

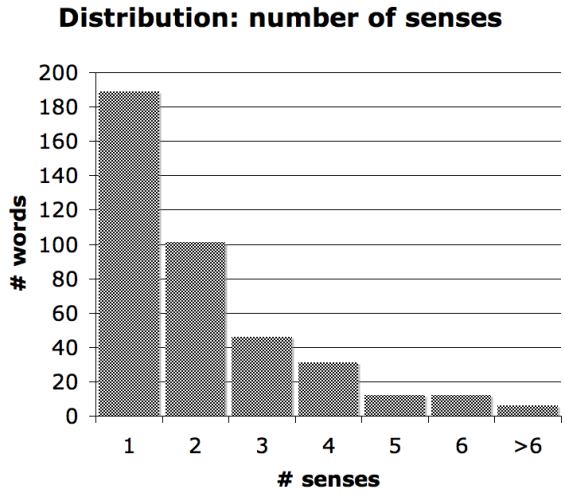


Figure 3: Distribution: number of words per number of senses. There are three words with seven senses, two words with nine senses and one word with ten senses. The average is 2.1 senses per word.

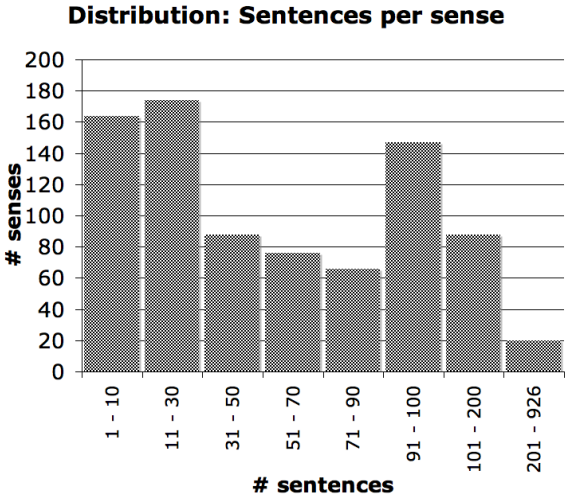


Figure 4: Distribution: number of senses per number of sentences interval. Only 5 senses have collected more than 300 sentences, at an average of 62.9 sentences per sense.

senses have a low number of sentences. Targets with only one sense have almost 100 sample sentences, resulting from the 100 sentences presented per iteration in Task 3. The experiment yielded a total of 51,736 sentences with a single sense-labeled target word. It is therefore built into our method to create a substantial corpus that could be used for training or evaluating WSD systems, complete with a level of frequency distribution among the senses, which is valuable in its own right. We collected substitutions for a total 8,771 sentences in Task 1. On average, a target word received 17 substitutions that were provided two or more times, and 4.5 substitutions with a frequency of ten or more. Manual inspection reveals that substitution frequencies over four are very reliable and virtually error-free.

5 Evaluation

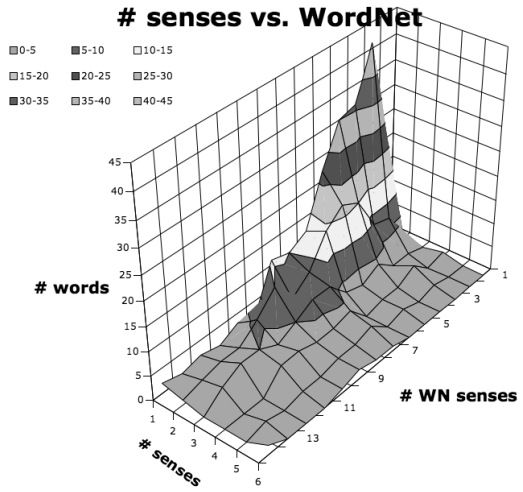
Having characterized the results of our experiment quantitatively, we now turn to assessing the quality of the results and compare the granularity of the results with WordNet.

5.1 Quantitive Comparison with WordNet

Since we developed this resource in order to overcome the excessive splitting of WordNet terms into senses, we now compare the granularity of our sense inventory with WordNet. For our 397 target words, WordNet 2.1 lists 2,448 senses (excluding named entity instances), an average of 6.17 senses per target and almost three times as many senses as listed in our inventory (average number of senses: 2.1). Looking at the data reveals that most fine-grained WordNet distinctions have been conflated into coarser-grained senses. Also, obscure WordNet senses have not been included in our inventory. On some occasions, our inventory lists more senses, e.g. the WordNet sense *station#n#1* includes both railway stations and gas stations, whereas the crowdsourcing process distinguishes these. Figure 5 provides a 3D-plot of the number of targets for most of the combinations of number of senses in WordNet and in our inventory. There is a direct correlation between the number of senses: words with a large number of WordNet senses also tend to be assigned a high number of senses in the crowdsourcing inventory.

5.2 Qualitative Evaluation

We now turn to the quality of the data obtained by our acquisition cycle. Since the goal of this resource is to provide sense labels and their substitutions, we are interested in how often the substitution assigned via the sense label is acceptable. Note that we only asked for substitutes on context for 8,771 sentences in Task 1, but project these substitutions via aggregation (clustering), inventory checking (Task 2) and matching the meaning (Task 3) to our full set of 51,736 sentences, which do not overlap with the sentences presented



X \ Y	1	2	3	4	5	6	≥ 7
1	20	1		1			
2	41	7	1				
3	31	8	1	3	1		
4	30	15	4	1			
5	22	17	2	1	1		
6	12	14	3				
7	10	8	7	4	3	1	
8	11	5	7	3	2	1	
9	2	10	5	3	2	2	
10	3	6	3	4	1	1	1
11	1	3	5	4	1		
12	2	3	5				1
13	1	2	3		1		1
14	2	1				2	1
≥ 15		1	3	4		5	2

Figure 5: Number of words that have X WordNet senses vs. Y senses in this inventory. 14 targets have 15 or more WordNet senses and are omitted.

in Task 1. This section describes an experiment to estimate the error rate of this projection. We selected the most frequent substitution per sense and added the second and third-ranked substitution in case their frequency was three or more. This produced three substitutions for most senses and one or two substitutions for minor senses. From our full list of sentences, we randomly sampled 500 sentences and set up an AMT task where we presented the sentence with the target word in bold along with a) the substitutions from this projection and b) random substitutions from the set of substitutions in separate tasks. Turkers were asked whether the substitutions matched the target word, matched the target word somewhat, do did match the target or the task was impossible for some reason. Table 1 shows the confusion matrix and the percentages of judgments, obtained by majority vote on five turkers. Manual checking of the positive answers for random substitutions revealed that

		System	
		Projection	Random
Vote	YES	469 (93.8%)	10 (2%)
	NO	14 (2.8%)	481 (96.2%)
	SOMEWHAT	17 (3.4%)	9 (1.8%)

Table 1: Evaluation of the substitute projection using majority vote on five turkers. Instances without majority are counted in the SOMEWHAT class.

these were in fact valid substitutions. For example, the substitutions of the municipality sense of ”community” were randomly selected for a sentence containing the municipality sense of ”city” as target.

Closer examination of the undecided and negative judgments for the projected substitutions showed that most of the negative judgments contained many judgments for ”somewhat matching” (whereas NO answers for randomly supplied substitutions where mostly unanimous). Other sources of negative judgments included minor senses that had only substitutions with frequency one. Given that less than 3% of projected substitutions were unanimously judged as not matching, while random substitutions were judged as not matching in over 96% of cases, we concluded that the data produced by our process is of very high quality and is suitable for both evaluation and training Word Sense Disambiguation systems.

5.3 Error Analysis

In this section we report findings on analyzing the bootstrapping process for a total of 100 target words. Systematic errors of the process can serve as indicators for improvements in later versions of out acquisition cycle.

For the 100 targets, we observed the following problems (multitude shown in brackets):

- (4) Overlap or containment of one sense in the other leads to matching with two classes. This can be sorted out by taking into consideration the confusion matrix of meaning matching (Task 3) and the similarity of senses as measured in Task 2. An indicator of this is the number of set aside sentences in Task 3.

- (3) Systematic part-of speech tagger errors. Especially prevalent with targets that are more frequent in the non-noun reading, such as ”back”. Turkers did not consistently

mark POS errors as impossible (although instructed). However, they reliably distinguished among senses. For example, "back in time" and "back in the yard" received separate inventory entries.

- (3) Conflation of senses. Despite differences in meaning, two senses (as perceived by us) had sufficient overlap in their substitutions to not get clustered apart, as it happened for "relationship" in the business and personal sense. This was detected by repeatedly getting a lot of "uncovered" judgments in Task 3 yet no new senses via the substitution step in the cycle.

- (2) Oscillation of senses. Differences in the subjective judgment of turkers caused the sense inventory to oscillate between grouping and distinguishing senses, such as "over the centuries" vs. "the 16th century". With a larger team of trusted turkers this oscillation became less of an issue since a more diverse crowd drove the process in one direction or another.

In total, we observed a successful process for about 90% of targets, with minor problems in the remainder that seldom led to noise in the data. The following issues relate to the power-law nature of word sense distributions (cf. (Kilgarriff, 2004)), which results in many minor senses:

- Minor senses in set aside sentences. When sampling a set of 100 sentences for Task 3, minor senses are likely to be set aside or not taken up by the clustering for lack of support. We observed this in eight targets in our analysis. While a larger sample mitigates this problem, for most applications, we are not interested in minor senses because of their low incidence and we thus do not view this as a problem. Inevitably, some minor senses in the domain did not make it into the sense inventory; however, the cases never represented more than 4% of sample sentences.

- Few substitutions for minor senses. Of the 834 senses distinguished in our experiment, 41 did not get any substitution with frequency ≥ 2 and 142 senses did not record a substitution frequency of four or more. A way to overcome few substitutions for minor senses is to simply ask for more substitutions in the style of Task 1 for the inventory sentence or for the matched sentences for a sense in question.

6 Conclusion and Future Work

In this paper we have demonstrated how to create a high quality semantic resource from scratch. Using Amazon Turk as a crowdsourcing platform and breaking down the problem of substitution and word sense inventory acquisition into three simple tasks, we were able to produce a rich semantic resource for semantic indexing for a comparatively low cost. Further contributions of this work are the definition of word senses along substitutability and the usage of a bootstrapping process on human input.

Compared to WordNet, which is the most commonly used inventory for word sense disambiguation, our resource has a much richer set of sample usages, a larger set of substitutions, fewer fine-grained distinctions and provides a corpus-based estimate on word sense distribution. Our method does not need pre-processing other than lemmatizing and POS tagging and can be directly applied to other domains or languages.

We have run a pilot study for verb sense acquisition with equally encouraging results. Further work can proceed along two lines: On the one hand, one can explore how to enrich the resource itself, e.g. by expanding the target set, acquisition of hypernyms or other relations in Task 1 style, for example, or by creating synsets of senses with the same substitutions that can substitute for each other. However, we are mainly interested in whether this resource is better suited to serve as a sense and substitution inventory for search applications.

Some sample data will be made available for download by the time of publication and is available at request.

Acknowledgements

We would like to thank Julian Richardson for fruitful discussions and the evaluation of initial experiments. We are indebted to thank our team of trusted turkers for performing our tasks so diligently. Thanks goes to Livia Polanyi for valuable comments on an earlier version of the paper.

References

Eneko Agirre and Philip Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer, July.

Chris Biemann. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-06*, New York, USA.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of HLT-NAACL 2006*, pages 57–60.

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word. In *In Proceedings of Text, Speech, Dialogue*, pages 1–9. Springer-Verlag.

Ioannis P. Klapaftis and Suresh Manandhar. 2008. Word sense induction using graphs of collocations. In *Proceedings of the 18th European Conference On Artificial Intelligence (ECAI-2008)*, Patras, Greece, July. IOS Press.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic, June. Association for Computational Linguistics.

Rada Mihalcea and Dan Moldovan. 2001. Automatic generation of a coarse grained WordNet. In *Proceedings of the NAACL worshop on WordNet and Other Lexical Resources*, Pittsburg, USA.

Rada Mihalcea. 1998. SEMCOR semantically tagged corpus. unpublished manuscript.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

Mark Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 142–151. ACM/Springer.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA*, pages 254–263.