

## Coarse Word-Sense Disambiguation Using Common Sense

**Catherine Havasi**  
MIT Media Lab  
havasi@media.mit.edu

**Robert Speer**  
MIT Media Lab  
rspeer@mit.edu

**James Pustejovsky**  
Brandeis University  
jamesp@cs.brandeis.edu

### Abstract

Coarse word sense disambiguation (WSD) is an NLP task that is both important and practical: it aims to distinguish senses of a word that have very different meanings, while avoiding the complexity that comes from trying to finely distinguish every possible word sense. Reasoning techniques that make use of common sense information can help to solve the WSD problem by taking word meaning and context into account. We have created a system for coarse word sense disambiguation using blending, a common sense reasoning technique, to combine information from SemCor, WordNet, ConceptNet and Extended WordNet. Within that space, a correct sense is suggested based on the similarity of the ambiguous word to each of its possible word senses. The general blending-based system performed well at the task, achieving an f-score of 80.8% on the 2007 SemEval Coarse Word Sense Disambiguation task.

### Common Sense for Word Sense Disambiguation

When artificial intelligence applications deal with natural language, they must frequently confront the fact that words with the same spelling can have very different meanings. The task of word sense disambiguation (WSD) is therefore critical to the accuracy and reliability of natural language processing. The problem of understanding ambiguous words would be greatly helped by understanding the relationships between the meanings of these words and the meaning of the context in which they are used – information that is largely contained in the domain of common sense knowledge.

Consider, for example, the word *bank* and two of its prominent meanings. In one meaning, a bank is a business institution where one would deposit money, cash checks, or take out loans: “The bank gave out fewer loans since the recession.” In the second, the word refers to the edges of land around the river, such as in “I sat by the bank with my grandfather, fishing.” We can use common sense to understand there would not necessarily be loans near a river, and rarely would fishing take place in a financial institution. We know that a money bank is different from a river bank because they have

different common-sense features, and those features affect the words that are likely to appear with the word *bank*.

In developing the word sense disambiguation process that we present here, our aim is to use an existing technique, called “blending” (Havasi et al. 2009), that was designed to integrate common sense into other applications and knowledge bases. Blending creates a single vector space that models semantic similarity and associations from several different resources, including common sense. We use generalized notions of similarity and association within that space to produce disambiguations. Using this process, instead of introducing a new and specialized process for WSD, will help to integrate disambiguation into other systems that currently use common sense.

### Coarse-Grained Word Sense Disambiguation

A common way to evaluate word sense disambiguation systems is to compare them to gold standards created by human annotators. However, many such corpora suffer low inter-annotator agreement: they are full of distinctions which are difficult for humans to judge, at least from the documentation (i.e. glosses) provided.

As a solution to this, the coarse word sense disambiguation (Coarse WSD) task was introduced by the SemEval evaluation exercise. In the coarse task, the number of word senses has been reduced. In Figure 1 we can see this simplification.

Coarse word senses allow for higher inter-annotator agreement. In the fine-grained Senseval-3 WSD task, there was an inner-annotator agreement of 72.5% (Snyder and Palmer 2004); this annotation used expert lexicographers. The Open Mind Word Expert task used untrained internet volunteers for a similar task<sup>1</sup> (Chklovski and Mihalcea 2002) and received an inter-annotator agreement score of 67.3%. These varying and low inter-annotator agreement scores call into question the relevance of fine-grained distinctions.

### The Coarse Grained Task

SemEval 2007 Task 7 was the “Coarse-Grained English All-Words Task” (Navigli and Litkowski 2007) which examines the traditional WSD task in a coarse-grained way, run by

<sup>1</sup>The Open Mind project is a family of projects started by David Stork, of which Open Mind Common Sense is a part. Thus Open Mind Word Expert is not a part of OMCS.

---

### Fine-grained

1. pen: pen (a writing implement with a point from which ink flows)
2. pen: pen (an enclosure for confining livestock)
3. pen: playpen, pen (a portable enclosure in which babies may be left to play)
4. pen: penitentiary, pen (a correctional institution for those convicted of major crimes)
5. pen: pen (female swan)

### Coarse-grained

1. pen: pen (a writing implement with a point from which ink flows)
  2. pen: pen (an enclosure – this contains the fine senses for livestock and babies)
  3. pen: penitentiary, pen (a correctional institution for those convicted of major crimes)
  4. pen: pen (female swan)
- 

Figure 1: The coarse and fine word senses for the word *pen*.

Roberto Navigli and Ken Litkowski (Navigli and Litkowski 2007). In the coarse task, the number of word senses has been dramatically reduced, allowing for higher inter-annotator agreement (Snyder and Palmer 2004; Chklovski and Mihalcea 2002).

Navigli and Litkowski tagged around 6,000 words with coarse-grained WordNet senses in a test corpus. They developed 29,974 coarse word senses for nouns and verbs, representing 60,655 fine WordNet senses; this is about a third of the size of the fine-grained disambiguation set. The senses were created semi-automatically using a clustering algorithm developed by the task administrators (Navigli 2006), and then manually verified. The coarse-grained word sense annotation task received inter-annotator agreement score of 86.4% (Snyder and Palmer 2004).

### Why Coarse Grained?

Although we have chosen to evaluate our system on the coarse-grained task, we believe common sense would help with any word sense disambiguation task for the reasons we described above. In this study, we have chosen coarse word sense disambiguation because of its alignment with the linguistic perceptions of the everyday people who built our crowd-sourced corpus of knowledge. We believe the coarse word sense task best aligns with the average person’s “common sense” of different word senses.

### The Semeval Systems

Fourteen systems were submitted to the Task 7 evaluation from thirteen different institutions (Navigli, Litkowski, and Hargraves 2007). Two baselines for this task were calculated. The first, the most frequent sense (MFS) baseline, performed

	cat	dog	airplane	toaster
... IsA pet	+6	+5		
... AtLocation home	+8	+2		+1
... CapableOf fly	-3	-5	+9	
... MadeOf metal			+1	+1
fur PartOf ...	+6	+5		

Figure 2: An example input matrix to AnalogySpace.

at 78.89% and the second, a random baseline, performed at 52.43%. The full results can be seen in Table 1 with the inclusion of our system’s performance.

We will examine the three top performing systems in more detail. The top two systems, NUS-PT and NUS-ML, were both from the National University of Singapore. The NUS-PT system (Chan, Ng, and Zhong 2007) used a parallel-text approach with a support vector learning algorithm. NUS-PT also used the SemCor corpus and the Defense Science Organization (DSO) disambiguated corpus. The NUS-ML system (Cai, Lee, and Teh 2007) focuses on clustering bag-of-words features using a hierarchical Bayesian LDA model. These features are learned from a locally-created collection of collocation features. These features, in addition to part-of-speech tags and syntactic relations, are used in a naïve Bayes learning network.

The LCC-WSD (Novischi, Srikanth, and Bennett 2007) system was created by the Language Computer Corporation. To create their features, they use a variety of corpora: SemCor, Senseval 2 and 3, and Open Mind Word Expert. In addition, they use WordNet glosses, eXtended WordNet, syntactic information, information on compound concepts, part-of-speech tagging, and named entity recognition. This information is used to power a maximum entropy classifier and support vector machines.

### Open Mind Common Sense

Our system is based on information and techniques used by the Open Mind Common Sense project (OMCS). OMCS has been compiling a corpus of common sense knowledge since 1999. It’s knowledge is expressed as a set of over one million simple English statements which tend to describe how objects relate to one another, the goals and desires people have, and what events and objects cause which emotions.

To make the knowledge in the OMCS corpus accessible to AI applications and machine learning techniques, we transform it into a semantic network called ConceptNet (Havasi, Speer, and Alonso 2007). ConceptNet is a graph whose edges, or *relations*, express common sense relationships between two short phrases, known as *concepts*. The edges are labeled from a set of named *relations*, such as *IsA*, *HasA*, or *UsedFor*, expressing what relationship holds between the concepts. Both ConceptNet and OMCS are freely available.

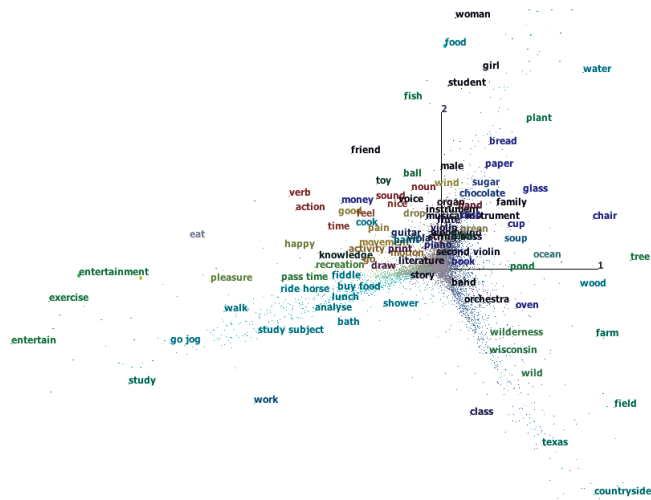


Figure 3: A projection of AnalogySpace onto two principal components, with some points labeled.

## AnalogySpace

AnalogySpace (Speer, Havasi, and Lieberman 2008) is a matrix representation of ConceptNet that is “smoothed” using dimensionality reduction. It expresses the knowledge in ConceptNet as a matrix of concepts and the common-sense features that hold true for them, such as “. . . is part of a car” or “a computer is used for . . .”. This can be seen in Figure 2.

Reducing the dimensionality of this matrix using truncated singular value decomposition has the effect of describing the knowledge in ConceptNet in terms of its most important correlations. A common operation that one can perform using AnalogySpace is to look up concepts that are similar to or associated with a given concept, or even a given set of concepts and features. A portion of the resulting space can be seen in Figure 3.

This is the kind of mechanism we need to be able to distinguish word senses based on their common sense relationships to other words, except for the fact that ConceptNet itself contains no information that distinguishes different senses of the same word. If we had a ConceptNet that knew about word senses, we could use the AnalogySpace matrix to look up which sense of a word is most strongly associated with the other nearby words.

## Blending

To add other sources of knowledge that *do* know about word senses (such as WordNet and SemCor) to AnalogySpace, we use a technique called “blending” (Havasi et al. 2009). Blending is a method that extends AnalogySpace, using singular value decomposition to integrate multiple systems or representations. Blending works by combining two or more data sets in the pre-SVD matrix, using appropriate weighting factors, to produce a vector space that represents correlations within and across all of the input representations.

Blending can be thought of as a way to use SVD-based reasoning to integrate common sense intuition into other data sets and tasks. Blending takes the AnalogySpace reasoning

process and extends them to work over multiple data sets, allowing analogies to propagate over different forms of information. Thus we can extend the AnalogySpace principle over different domains: other structured resources, free text, and beyond. Blending requires only a rough alignment of resources in its input, allowing the process to be quick, flexible and inclusive.

The motivation for blending is simple: you want to combine multiple sparse-matrix representations of data from different domains, essentially by aligning them to use the same labels and then summing them. But the magnitudes of the values in each original data set are arbitrary, while their relative magnitudes when combined make a huge difference in the results. We want to find relative magnitudes that encourage as much interaction as possible between the different input representations, expanding the domain of reasoning across all of the representations. Blending heuristically suggests how to weight the inputs so that this happens, and this weight is called the blending factor.

## Bridging

To make blending work, there has to be some overlap in the representations to start with; from there, there are strategies for developing an optimal blend (Havasi 2009). One useful strategy, called “bridging”, helps create connections in an AnalogySpace between data sets that do not appear to overlap, such as a disambiguated resource and a non-disambiguated resource.

A third “bridging” data set may be used to create overlap between the data sets (Havasi, Speer, and Pustejovsky 2009). An example of this is making a connection between WordNet, whose terms are disambiguated and linked together through synsets, and ConceptNet, whose terms are not disambiguated. To bridge the data sets, we include a third data set that we call “Ambiguated WordNet”, which expresses the connections in WordNet with the terms replaced by ambiguous terms that line up with ConceptNet.

## Blending Factors

Next, we calculate weight factors for the blend, by comparing the top singular values from the various matrices. Using those values, we choose the blending factor so that the contributions of each matrix’s most significant singular value are equal. This is the “rough blending heuristic”, as described in Havasi (Havasi 2009).

We can blend more than two data sets by generalizing the equation for two data sets, choosing a set of blending factors such that each pair of inputs has the correct relative weight. This creates a reasoning AnalogySpace which is influenced by each matrix equally.

The blend used for this task is a complex blend of multiple sources of linguistic knowledge, both ambiguous and disambiguated, such as Extended WordNet, SemCor, and ConceptNet. We will discuss its creation below.

## Methodology for Disambiguation

Here, we set up a blending-based system to perform sparse word sense disambiguation. In this system, we used blending to create a space representing the relations and contexts surrounding both disambiguated and ambiguous words, those without attached word sense encodings.

We can use this space to discover which word sense an ambiguous word is most similar to, thus disambiguating the word in question. We can discover similarity by considering dot products, providing a measure that is like cosine similarity but is weighted by the magnitudes of the vectors.

This measure is not strictly a similarity measure, because identical vectors do not necessarily have the highest possible dot product. It can be considered, however, to represent the *strength* of the similarity between the two vectors, based on the amount of information known about them and their likelihood of appearing in the corpus. Pairs of vectors, each vector representing a word in this space, have a large dot product when they are frequently used and have many semantic features in common.

To represent the expected semantic value of the sentence as a whole, we can average together the vectors corresponding to all words in the sentence (in their ambiguous form). The resulting vector does not represent a single meaning; it represents the “ad hoc category” (Havasi, Speer, and Pustejovsky 2009) of meanings that are similar to the various possible meanings of words in the sentence. Then, to assign word senses to the ambiguous words, we find the sense of each word that has the highest dot product (and thus the strongest similarity) with the sentence vector.

A simple example of this process is shown in Figure 5. Suppose we are disambiguating the sentence “I put my money in the bank”. For the sake of simplicity, suppose that there are only two possible senses of “bank”:  $bank_1$  is the institution that stores people’s money, and  $bank_2$  is the side of a river.

The three content words, “put”, “money”, and “bank”, each correspond to a vector in the semantic space. The sentence vector,  $S$ , is made from the average of these three. The two senses of “bank” also have their own semantic vectors. To choose the correct sense, then, we simply calculate that  $bank_1$  has a higher dot product with  $S$  than  $bank_2$  does, indicating

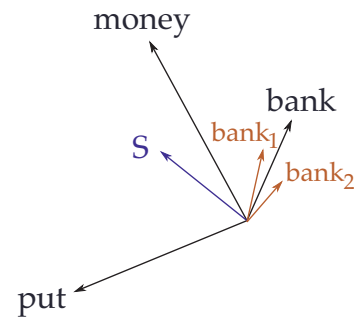


Figure 5: An example of disambiguation on the sentence “I put my money in the bank”.

that it is the most likely to co-occur with the other words in the sentence.

This is a simplified version of the actual process, and it makes the unnecessary assumption that all the words in a sentence are similar to each other. As we walk through setting up the actual disambiguation process, we will create a representation that is more applicable for disambiguation, because it will allow us to take into account words that are not directly similar to each other but *are* likely to appear in the same sentence.

## The Resources

First, we must create the space that we use in these calculations. To do so, we must choose resources to include in the blended space. These resources should create a blend with knowledge about the senses in WordNet and add knowledge from ConceptNet, so we can distinguish word senses based on their common-sense features. Additionally, we want to add the information in SemCor, the gold standard corpus that is the closest match to a “training set” for SemEval. Whenever we make a blend, we need to ensure that the data overlaps, so that knowledge can be shared among the resources.

In the blend we used: ConceptNet 3.5 in its standard matrix form; WordNet 3.0, expressed as relations between word senses; a “pure association” matrix of ConceptNet 3.5, describing only that words are connected and without distinguishing which relation connects them; an “ambiguous” version of WordNet 3.0, which creates alignment with ConceptNet by not including sense information; eXtended WordNet (XWN), which adds more semantic relations to WordNet 2.0 that are extracted from each entry’s definition; “ambiguous” versions of Extended WordNet; and the “brown1” and “brown2” sections of SemCor 3.0, as an association matrix describing which words or word senses appear in the same sentence, plus their “ambiguous” versions.

## Aligning the Resources

To share information between different sources, blending requires overlap between their concepts or their features, but blending does not require all possible pairs of resources to overlap. One obstacle to integrating these different resources was converting their different representations of

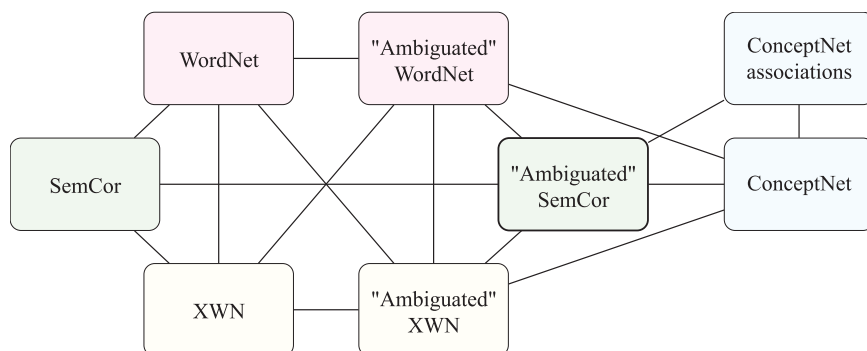


Figure 4: A diagram of the blend we use for word sense disambiguation. Resources are connected when they have either concepts or features in common.

WordNet senses and parts of speech to a common representation. Because SemEval is expressed in terms of WordNet 2.1 senses, we converted all references to WordNet senses into WordNet 2.1 sensekeys using the conversion maps available at <http://wordnet.princeton.edu/wordnet/download/>.

As this was a coarse word sense disambiguation task, the test set came with a mapping from many WordNet senses to coarse senses. For the words that were part of a coarse sense, we replaced their individual sensekeys with a common identifier for the coarse sense.

For the purpose of conserving memory usage, when we constructed matrices representing the relational data in WordNet, we discarded multiple-word collocations. The matrices only represented WordNet entries containing a single word.

To maximize the overlap between resources, we added the alternate versions of some resources that are listed above. One simple example is that in addition to ConceptNet triples such as (dog, CapableOf, bark), we also included “pure association” relations such as (dog, Associated, bark).

The data we collect from SemCor also takes the form of pure associations. If the sense  $car_1$  and the sense  $drive_2$  appear in a sentence, for example, we will give  $car_1$  the feature *associated/drive<sub>2</sub>* and give  $drive_2$  the feature *associated/car<sub>1</sub>*.

Given a disambiguated resource such as WordNet or SemCor, we also needed to include versions of it that could line up with ambiguous resources such as ConceptNet or the actual SemEval test data. The process we call “ambiguation” replaces one or both of the disambiguated word senses, in turn, with ambiguous versions that are run through ConceptNet’s lemmatizer. An example is given below:

Given the disambiguated triple ( $sense1$ ,  $rel$ ,  $sense2$ ):

- Add the triple ( $amb1$ ,  $rel$ ,  $amb2$ ) (where  $amb1$  and  $amb2$  are the ambiguous, lemmatized versions of  $sense1$  and  $sense2$ ).
- Add the triple ( $amb1$ ,  $rel$ ,  $sense2$ ).
- Add the triple ( $sense1$ ,  $rel$ ,  $amb2$ ).

Blending works through shared information. Figure 4 shows the components of the blend and identifies the ones that share information with each other. The “ambiguated”

SemCor, which occupies a fairly central position in this diagram, contains the same type of information as the ambiguous texts which are part of the SemEval evaluation.

### Disambiguation using the Blend

Now that we have combined the resources together into a blended matrix, we must use this matrix to disambiguate our ambiguous words. For each sentence in our disambiguated test corpus, we create an ad hoc category representing words and meanings that are likely to appear in the sentence. Instead of simply averaging together the vectors for the terms, we average the features for things that have the “associated” relation with those terms. This is the new relation that we created above and used with SemCor and ConceptNet.

Consider again the sentence *I put my money in the bank*. We look for words that are likely to carry semantic content, and extract the non-stopwords *put*, *money*, and *bank*. From them we create the features: *associated/put*, *associated/money*, and *associated/bank*, and average those features to create an ad hoc category of “word meanings that are associated with the words in this sentence”.

For each word that is to be disambiguated, we find the sense of the word whose vector that has the highest dot product with the ad hoc category’s vector. If no sense has a similarity score above zero, we fall back on the most common word sense for that word.

It is important not to normalize the magnitudes of the vectors in this application. By preserving the magnitudes, more common word senses get larger dot products in general. The disambiguation procedure is thus considerably more likely to select more common word senses, as it should be: notice that the simple baseline of choosing the most frequent sense performs better than many of the systems in Task 7 did.

### SemEval Evaluation

The SemEval 2007 test set for coarse word sense disambiguation contains five documents in XML format. Most content words are contained in a tag that assigns the word a unique ID, and gives its part of speech and its WordNet lemma. The goal is to choose a WordNet sense for each tagged word so that it matches the gold standard.

System	F1
NUS-PT	82.50
NUS-ML	81.58
LCC-WSD	81.45
<b>Blending</b>	80.8
GPLSI	79.55
$BL_{MFS}$	78.89
UPV-WSD	78.63
SUSSX-FR	77.04
TKB-UO	70.21
PU-BCD	69.72
RACAI-SYNWSD	65.71
SUSSX-C-WD	64.52
SUSSX-CR	64.35
USYD	58.79
UoFL	54.61
$BL_{rand}$	52.43

Table 1: Task 7 systems scores sorted by F1 measure, including the performance of our blending-based system.

Our disambiguation tool provided an answer for 2262 of 2269 words. (The remaining seven words produced errors because our conversion tools could not find a WordNet entry with the given lemma and part of speech.) 1827 of the answers were correct, giving a precision of  $1827/2262 = 80.8\%$  and a recall of  $1827/2269 = 80.5\%$ , for an overall F-score of 80.6%. The blending-based system is compared to the other SemEval systems in Table 1.

When the results for SemEval 2007 were tallied, the organizers allowed the algorithms to fall back on a standard list of the most frequent sense of each word in the test set, in the cases where they did not return an answer. This improved the score of every algorithm with missing answers. Applying this rule to our seven missing answers makes a slight difference in our F-score, raising it to 80.8%.

Even though prediction using blending and ad hoc categories is a general reasoning tool that is not fine-tuned for the WSD task, this score would put us at fourth place in the SemEval 2007 rankings, as shown in Table 1.

## Future Work

The results of this paper show promise for the use of general common sense based techniques such as blending. We're interested in continuing to apply common sense to linguistic tasks, perhaps prepositional phrase attachment.

In the future it would be interesting to explore a fine-grained word sense task, perhaps in a different language. The OMCS project has been extended to other languages, with sites in Portuguese, Chinese, Korean, and Japanese. These languages could also serve as parallel corpora for a more advanced word sense disambiguation system.

## References

Cai, J. F.; Lee, W. S.; and Teh, Y. W. 2007. Nus-ml: Improving word sense disambiguation using topic features. In *Proceedings of the 4th International Workshop on Semantic*

*Evaluations (SemEval)*, 249–252. Prague, Czech Republic: Association for Computational Linguistics.

Chan, Y. S.; Ng, H. T.; and Zhong, Z. 2007. NUS-PT: Exploiting parallel texts for word sense disambiguation in the English all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, 253–256. Prague, Czech Republic: Association for Computational Linguistics.

Chklovski, T., and Mihalcea, R. 2002. Building a sense tagged corpus with Open Mind Word Expert. In *Proceedings of the ACL-02 workshop on Word sense disambiguation*, 116–122. Morristown, NJ, USA: Association for Computational Linguistics.

Havasi, C.; Speer, R.; Pustejovsky, J.; and Lieberman, H. 2009. Digital intuition: Applying common sense using dimensionality reduction. *IEEE Intelligent Systems*.

Havasi, C.; Speer, R.; and Alonso, J. 2007. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*.

Havasi, C.; Speer, R.; and Pustejovsky, J. 2009. Automatically suggesting semantic structure for a generative lexicon ontology. In *Proceedings of the Generative Lexicon Conference*.

Havasi, C. 2009. *Discovering Semantic Relations Using Singular Value Decomposition Based Techniques*. Ph.D. Dissertation, Brandeis University.

Navigli, R., and Litkowski, K. C. 2007. SemEval-2007: Task Summary. SemEval Web site. <http://nlp.cs.swarthmore.edu/semeval/tasks/task07/summary.shtml>.

Navigli, R.; Litkowski, K. C.; and Hargraves, O. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, 30–35. Prague, Czech Republic: Association for Computational Linguistics.

Navigli, R. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 105–112. Morristown, NJ, USA: Association for Computational Linguistics.

Novischi, A.; Srikanth, M.; and Bennett, A. 2007. LCC-WSD: System description for English coarse grained all words task at SemEval 2007. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, 223–226. Prague, Czech Republic: Association for Computational Linguistics.

Snyder, B., and Palmer, M. 2004. The English all-words task. In Mihalcea, R., and Edmonds, P., eds., *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 41–43. Barcelona, Spain: Association for Computational Linguistics.

Speer, R.; Havasi, C.; and Lieberman, H. 2008. AnalogySpace: Reducing the dimensionality of common sense knowledge. *Proceedings of AAAI 2008*.