

# Refining the most frequent sense baseline

**Judita Preiss**

Department of Linguistics  
The Ohio State University  
judita@ling.ohio-state.edu

**Jon Dehdari**

Department of Linguistics  
The Ohio State University  
jonsafari@ling.ohio-state.edu

**Josh King**

Computer Science and Engineering  
The Ohio State University  
kingjo@cse.ohio-state.edu

**Dennis Mehay**

Department of Linguistics  
The Ohio State University  
mehay@ling.ohio-state.edu

## Abstract

We refine the most frequent sense baseline for word sense disambiguation using a number of novel word sense disambiguation techniques. Evaluating on the SENSEVAL-3 English all words task, our combined system focuses on improving every stage of word sense disambiguation: starting with the lemmatization and part of speech tags used, through the accuracy of the most frequent sense baseline, to highly targeted individual systems. Our supervised systems include a ranking algorithm and a Wikipedia similarity measure.

## 1 Introduction

The difficulty of outperforming the most frequent sense baseline, the assignment of the sense which appears most often in a given annotated corpus, in word sense disambiguation (WSD) has been brought to light by the recent SENSEVAL WSD system evaluation exercises. In this work, we present a combination system, which, rather than designing a single approach to all words, enriches the most frequent sense baseline when there is high confidence for an alternative sense to be chosen.

WSD, the task of assigning a sense to a given word from a sense inventory is clearly necessary for other natural language processing tasks. For example, when performing machine translation, it is necessary to distinguish between word senses in the original language if the different senses have different possible translations in the target language (Yngve, 1955). A number of different approaches to WSD have been explored in recent years, with two

distinct approaches: techniques which require annotated training data (supervised techniques) and techniques which do not (unsupervised methods).

It has long been believed that supervised systems, which can be tuned to a word's context, greatly outperform unsupervised systems. This theory was supported in the SENSEVAL WSD system evaluation exercises, where the performance gap between the best supervised system and the best unsupervised system is large. Unsupervised systems were found to never outperform the most frequent sense (MFS) baseline (a sense assignment made on the basis of the most frequent sense in an annotated corpus), while supervised systems occasionally perform better than the MFS baseline, though rarely by more than 5%. However, recent work by McCarthy et al. (2007) shows that acquiring a predominant sense from an unannotated corpus can outperform many supervised systems, and under certain conditions will also outperform the MFS baseline.

Rather than proposing a new algorithm which will tackle all words, we focus on improving upon the MFS baseline system when an alternative system proposes a high confidence answer. An MFS refining system can therefore benefit from answers suggested by a very low recall (but high precision) WSD system. We propose a number of novel approaches to WSD, but also demonstrate the importance of a highly accurate lemmatizer and part of speech tagger to the English all words task of SENSEVAL-3.<sup>1</sup>

We present our enriched most frequent sense

---

<sup>1</sup>Unless specified otherwise, we use WordNet 1.7.1 (Miller et al., 1990) and the associated sense annotated SemCor corpus (Miller et al., 1993) (translated to WordNet 1.7.1 by Rada Mihalcea).

baseline in Section 2, which motivates the lemmatizer and part of speech tagger refinements presented in Section 3. Our novel high precision WSD algorithms include a reranking algorithm (Section 4), and a Wikipedia-based similarity measure (Section 5). The individual systems are combined in Section 6, and we close with our conclusions in Section 7.

## 2 Most frequent sense baseline

The most frequent sense (MFS) baseline assumes a sense annotated corpus from which the frequencies of individual senses are learnt. For each target word, a part of speech tagger is used to determine the word’s part of speech, and the MFS for that part of speech is selected. Although this is a fairly naive baseline, it has been shown to be difficult to beat, with only 5 systems of the 26 submitted to the SENSEVAL-3 English all words task outperforming the reported 62.5% MFS baseline. The success of the MFS baseline is mainly due to the frequency distribution of senses, with the shape of the sense rank versus frequency graph being a Zipfian curve (i.e., the top-ranked sense being much more likely than any other sense).

However, two different MFS baseline performance results are reported in Snyder and Palmer (2004), with further implementations being different still. The differences in performance of the MFS baseline can be attributed to a number of factors: the English all words task is run on natural text and therefore performance greatly depends on the accuracy of the lemmatizer and the part of speech tagger employed.<sup>2</sup> If the lemmatizer incorrectly identifies the stem of the word, the MFS will be looked up for the wrong word and the resulting sense assignment will be incorrect. The performance of the MFS given the correct lemma and part of speech information is 66%, while the performance of the MFS with a Port Stemmer without any POS information is 32%. With a TreeTagger (Schmidt, 1994), and a sophisticated lemma back-off strategy, the performance increases to 56%. It is this difference in

<sup>2</sup>Other possible factors include: 1) The sense distribution in the corpus which the MFS baseline is drawn from, 2) If SemCor is used as the underlying sense annotated corpus, the accuracy of the mapping from WordNet 1.6 (with which SemCor was initially annotated) to WordNet 1.7.1 could also have an effect on the performance).

performance which motivates refining the most frequent sense baseline, and our work on improving the underlying lemmatizer and part of speech tagger presented in Section 3.

Our initial investigation refines the SemCor based MFS baseline using the automatic method of determining the predominant sense presented in McCarthy et al. (2007).

1. For nouns and adjectives which appear in SemCor fewer than 5 times, we employ the automatically determined predominant sense.
2. For verbs which appear in SemCor fewer than 5 times, we employ subcategorization frame similarity rather than Lesk similarity to give us a verb’s predominant sense.

### 2.1 Predominant sense

McCarthy et al. (2007) demonstrate that it is possible to acquire the predominant sense for a word in a corpus without having access to annotated data. They employ an automatically created thesaurus (Lin, 1998), and a sense–word similarity metric to assign to each sense  $s_i$  of a word  $w$  a score corresponding to

$$\sum_{n_j \in N_w} dss(w, n_j) * \frac{sss(s_i, n_j)}{\sum_{s'_i \in \text{senses}(w)} sss(s'_i, n_j)}$$

where  $dss(w, n_j)$  reflects the distributional similarity of word  $w$  to  $n_j$ ,  $w$ ’s thesaural neighbour, and  $sss(s_i, n_j) = \max_{s_x \in \text{senses}(n_j)} sss'(s_i, s_x)$  is the maximum similarity<sup>3</sup> between  $w$ ’s sense  $s_i$  and a sense  $s_x$  of  $w$ ’s thesaural neighbour  $n_j$ . The authors show that although this method does not always outperform the MFS baseline based on SemCor, it does outperform it when the word’s SemCor frequency is below 5. We therefore switch our MFS baseline to this value for such words. This result is represented as ‘McCarthy’ in Table 1, which contains the results of the techniques presented in this Section evaluated on the SENSEVAL-3 English all words task.

### 2.2 Verb predominant sense

McCarthy et al. (2007) observe that their predominant sense method is not performing as well for

<sup>3</sup>We use the Lesk (overlap) similarity as implemented by the WordNet::similarity package (Pedersen et al., 2004).

System	Precision	Recall	<i>F</i> -measure
MFS	58.4%	58.4%	58.4%
McCarthy	58.5%	58.5%	58.5%
Verbs	58.5%	58.5%	58.5%
All	58.6%	58.6%	58.6%

Table 1: Refining the MFS baseline with predominant sense

verbs as it does for nouns and adjectives. We hypothesize that this is due to the thesaural neighbours obtained from Lin’s thesaurus, and we group verbs according to the subcategorization frame (SCF) distributions they present in the VALEX (Korhonen et al., 2006) lexicon. A word  $w_1$  is grouped with word  $w_2$  if the Bhattacharyya coefficient

$$BC(w_1, w_2) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

where  $p(x)$  and  $q(x)$  represent the probability values for subcategorization class  $x$ , is above a certain threshold. The *BC* coefficient then replaces the *ds* value in the original formula and the predominant senses are obtained. Again, this system is only used for words with frequency lower than 5 in SemCor. The great advantage of the Bhattacharyya coefficient over various entropy based similarity measures which are usually used to compare SCF distributions (Korhonen and Krymolowski, 2002), is that it is guaranteed to lie between 0 and 1, unlike the entropy based measures which are not easily comparable between different word pairs. This result is represented by ‘Verbs’ in Table 1.

Table 1 displays the results for the MFS, the MFS combined with the two approaches described above, and the MFS combining MFS with verbs and McCarthy.

### 3 Lemmatization and Part of Speech Tagging

We made use of several lemmatizers and part-of-speech taggers, in order to give the other WSD components the best starting point possible.

#### 3.1 Lemmatization

Lemmatization, the process of obtaining the canonical form of a word, was the first step for us to ultimately identify the correct WordNet sense of

a given word in the English all words task. We found that without any lemmatizing of the test input, the maximum *f*-score possible was in the mid-50’s. Conversely, we found that a basic most-frequent-sense system that had a perfectly-lemmatized input achieved an *f*-score in the mid-60’s. This large difference in the ceiling of a non-lemmatized system and the floor of a perfectly-lemmatized system motivated us to focus on this task.

We looked at three different lemmatizers: the lemmatizing backend of the XTAG project (XTAG Research Group, 2001)<sup>4</sup>, Celex (Baayen et al., 1995), and the lemmatizing component of an enhanced TBL tagger (Brill, 1992).<sup>5</sup> We then employed a voting system on these three components, taking the lemma from the most individual lemmatizers. If all three differ, we take the lemma from the most accurate individual system, namely the TBL tagger.

#### 3.1.1 Lemmatizer Evaluation

We evaluated the lemmatizers against the lemmas found in the SENSEVAL-3 gold standard.<sup>6</sup> Even the lowest performing system improved accuracy by 31.74% over the baseline, which baseline simply equates the given token with the lemma. Table 2 shows the results of evaluating the lemmatizers against the EAW key.

While the simple voting system performed better than any of the individual lemmatizers, hyphenated words proved problematic for all of the systems. Some hyphenated words in the test set remained hyphenated in the gold standard, and some others were separated. However, evaluation results show that splitting hyphenated words increases lemmatizing accuracy by 0.9% .

#### 3.2 Part of Speech Tagging

We also investigated the contribution of part of speech taggers to the task of word sense disambiguation. We considered three taggers: the Elworthy bigram tagger (Elworthy, 1994) within the RASP parser (Briscoe et al., 2006), an enhanced

<sup>4</sup><http://www.cis.upenn.edu/~xtag>

<sup>5</sup><http://gpostt1.sourceforge.net>

<sup>6</sup>We removed those lines from both the test input and the gold standard which were marked U (= unknown, 34 lines), and we removed the 40 lines from the test input that were missing from the gold standard. This gave us 2007 words in both the test set and the gold standard.

Lemmatizer	Accuracy
Baseline	57.50%
XTAG	89.24%
Celex	91.58%
TBL	92.38%
Voting {XTAG,Celex,TBL}	93.77%
Voting, no hyphen {XTAG,Celex,TBL}	<b>94.67%</b>

Table 2: Accuracy of several lemmatizers on <head> words of EAW task.

TBL tagger (Brill, 1992)<sup>7</sup> and a TnT-style trigram tagger (Halácsy et al., 2007).<sup>8</sup> The baseline was a unigram tagger which selects the most frequently-occurring tag of singletons when dealing with unseen words.

All three of the main taggers performed comparably, although only the Elworthy tagger provides probabilities associated with tags, rather than getting a single tag as output. This additional information can be useful, since we can employ different strategies for a word with one single tag with a probability of 1, versus a word with multiple tags, the most probable of which might only have a probability of 0.3 for example. For comparative purposes, we mapped the various instantiations of tags for nouns, verbs, adjectives, and adverbs to these four basic tags, and evaluated the taggers’ results against the EAW key. Table 3 shows the results of this evaluation.

The performance of these taggers on the EAW <head>-words is lower than results reported on other datasets. This can be explained by the lack of frequently-occurring function words, which are easy to tag and raise overall accuracy. Also, the words in the test set are often highly ambiguous not only with respect to their word sense, but also their part of speech.

#### 4 Supervised Learning of Sparse Category Indices for WSD

In this component of our refinement of the baseline, we train a supervised system that performs higher-precision classification, only returning an answer when a predictive feature that strongly predicts a particular sense is observed. To achieve this,

<sup>7</sup><http://gpostt1.sourceforge.net>

<sup>8</sup><http://code.google.com/p/hunpos>

POS Tagger	Accuracy
Baseline	84.10%
TBL	90.48%
Elworthy	90.58%
TnT	91.13%
Voting {TBL,Elw.,TnT}	<b>91.88%</b>

Table 3: Accuracy of several POS taggers on <head> words of EAW task.

we implemented a “feature focus” classifier (sparse weighted index) as described in (Madani and Connor, 2008, henceforth, MC08). MC08’s methods for restricting and pruning the number of feature-to-class associations are useful for finding and retaining only strong predictive features. Moreover, this allowed us to use a rich feature set (more than 1.6 million features) without an unwieldy explosion in the number of parameters, as feature-class associations that are not strong enough are simply dropped.

#### 4.1 Sparse Category Indices

MC08 describe a space and time efficient method for learning discriminative classifiers that rank large numbers of output classes using potentially millions of features for many instances in potentially tera-scale data sets. The authors describe a method for learning ‘category indices’ — i.e., weighted bipartite graphs  $G \subseteq F \times W \times C$ , where  $F$  is the set of features,  $C$  is the set of output classes and all weights (or ‘associations’)  $w \in W$  between features and the output classes they predict are real-valued and in  $[0.0, 1.0]$ . The space and time efficiency of MC08’s approach stems chiefly from three (parameterisable) restrictions on category indices and how they are updated. First, at any time in the learning process, only those edges  $(f_i, w_j, c_k) \in G$  whose associations  $w_j$  are a large enough proportion of the sum of all class associations for  $f_i$  are retained: that is, only retain  $w_j$  s.t.  $w_j \geq \mathbf{wmin}$ .<sup>9</sup> Second, by setting an upper bound  $\mathbf{dmax}$  on the number of associations that a feature  $f_i$  is allowed to have, only the largest feature associations are retained. Setting  $\mathbf{dmax}$  to a low number ( $\leq 25$ ) makes each feature a high-precision, low-recall predictor of output classes. Further, the  $\mathbf{dmax}$  and  $\mathbf{wmin}$  restrictions on parameter reten-

<sup>9</sup>Recall that  $w_j \in W$  are all between 0.0 and 1.0 and sum to 1.0.

tion allow efficient retrieval and update of feature weights, as only a small number of feature weights need be consulted for predicting output classes or learning from prediction mistakes in an online learning setting.<sup>10</sup> Finally, in the online learning algorithm,<sup>11</sup> in addition to the small number of features that need be consulted or updated, an error margin **marg** can be set so that parameter update only occurs when the  $\text{score}(c) - \text{score}(c^*) \leq \text{marg}$ , where  $c$  is the correct output class and  $c^* \neq c$  is the most confident incorrect prediction of the classifier. Setting **marg** = 0.0 leads to purely error-driven learning, while **marg** = 1.0 always updates on every learning instance. Values of **marg**  $\in$  (0.0, 1.0) will bias the category index learner to update at different levels of separation of the correct class from the most confident incorrect class, ranging from almost always error driven (near 0.0) to almost error-insensitive learning (near 1.0).

## 4.2 Integration into the WSD Task

Using both the Semcor-3 and English Lexical Sample training data sets (a total of  $\approx 45,000$  sentences, each with one or more labeled instances), we trained a sparse category index classifier as in MC08 with the following features: using words, lemmas and parts of speech (POSS) as tokens, we define features for (1) preceding and following unigrams and bigrams over tokens, as well as (2) the conjunction of the preceding unigrams (i.e., a 3-word window minus the current token) and (3) the conjunction of the preceding and following bigrams (5-word window minus the current token). Finally all surrounding lemmas in the sentence are treated as left- or right-oriented slot-independent features with an exponentially decaying level of activation  $\text{act}(l_i) = 0.5 \cdot \exp(0.5 \cdot -\text{dist}(l_i, \text{targ\_wd}))$  — where  $\text{dist}(l_i, \text{targ\_wd})$  is simply the word distance from the target word to the contextual lemma  $l_i$ .<sup>12</sup> Although WSD is not a many-class, large-

<sup>10</sup>**dmax** bounds the number of feature-class associations (parameters) must be consulted in prediction and updating, but, because of the **wmin** restriction, MC08 found that, on average, many fewer feature associations —  $\leq 16$  — were ever touched per training or testing instance in their classification experiments. See Madani and Connor (2008) for more details.

<sup>11</sup>Again, see Madani and Connor (2008) for more details.

<sup>12</sup>The value 0.5 is also a parameter that we have fixed, but it could in principle be tuned to a particular data set. In the interest of simplicity, we have not done this.

scale classification task,<sup>13</sup> we nevertheless found MC08’s pruning mechanisms useful for removing weak feature-word associations. Due to the aggressive pruning of feature-class associations, our model only has  $\approx 1.9\text{M}$  parameters out of a potential  $1,600,000 \times 200,000 = 320$  billion (the number of features times the number of WordNet 3.0 senses).

## 4.3 Individual System Results

To integrate the predictions of the classifier into the EAW task, we looked up all senses for each lemma-POS pairing, backing off to looking up the words themselves by the same POS, and finally resorting to splitting hyphenated words and rejoining multi-word units (as marked up in the EAW test set). Being high precision, the classifier does not return a valid answer for every lemma, so we report results with and without backing off to the most frequent sense baseline to fill in these gaps.

Individual system scores are listed in Table 4. The classifier on its own returns very few answers (with a coverage — as distinct from recall — of only 10.4% of the test set items). Although the classifier-only performance does not have broad enough coverage for stand-alone use, its predictions are nonetheless useful in combination with the baseline. Further, we expect coverage to grow when trained over a larger corpus (such as the very large web-extracted corpus of Agirre et al. (2004), which this learning method is well suited for).

## 5 Wikipedia for Word Sense Disambiguation

Wikipedia, an online, user-created encyclopedia, can be considered a collection of articles which link to each other. While much information exists within the textual content of Wikipedia that may assist in WSD, the approach presented here instead uses the article names and link structure within Wikipedia to find articles which are most related to a WordNet sense or context. We use the Green method to find a relatedness metric for articles from Wikipedia<sup>14</sup> (Ol-

<sup>13</sup>Large-scale data sets are available, but this does not change the level of polysemy in WordNet, which is not in the thousands for any given lemma.

<sup>14</sup>Computations were performed using a January 3<sup>rd</sup> 2008 download of the English Wikipedia.

Back-off	Precision	Recall	Prec. (n-best)	Rec. (n-best)
YES	0.592	0.589	0.594	0.589
No	0.622	0.065	0.694	0.070

Table 4: Precision and recall of sparse category index classifier — both “soft” scores of standard Senseval script and scores where any correct answer in list returned by the classifier is counted as a correct answer (‘n-best’). ‘Back-off’ signals whether the system backs off to the most frequent sense baseline.

livier and Senellart, 2007) based on each sense or context of interest.

Advantages of this method over alternative methods that attempt to incorporate Wikipedia into WSD is that our system is unsupervised and that no manual mapping needs to take place between WordNet and Wikipedia. Mihalcea (2007) demonstrates that manual mappings can be created for a small number of words with relative ease, but for a very large number of words the effort involved in mapping would approach presented involves no be considerable. The approach presented here involves no mapping between WordNet and Wikipedia but human effort in mapping between WordNet and Wikipedia, but instead initializes the Green method with a vector based only on the article names (as described in Section 5.2).

### 5.1 Green Method

The Green method (Ollivier and Senellart, 2007) is used to determine the importance of one node in a directed graph with respect to other nodes.<sup>15</sup> In the context of Wikipedia the method finds the articles which are most likely to be frequented if a random walk were used to traverse the articles, starting with a specific article and returning to that article if the random walk either strays too far off topic or to an article which is generally popular even without the context of the initial article. One of the features of the Green method is that it does not simply reproduce the global PageRank (Brin and Page, 1998), instead determining the related pages nearby due to relevance to the initial node.

The probability that the random walker of Wikipedia will transfer to an article is defined as a uniform distribution over the outlinks of the page where the random walker is currently located. As an approximation to the method described by Ol-

<sup>15</sup>In subsequent sections we give a high-level description of using the Green method with Wikipedia, however see Ollivier and Senellart (2007) for a much more detailed explanation.

livier and Senellart (2007), we create a subgraph of Wikipedia for every computation, comprised of the articles within a distance of 2 outlink traversals from the initial articles. Since Wikipedia is very highly connected, this constructed subgraph still contains a large number of articles and performance of the Green method on this subgraph is similar to that on the whole connectivity graph.

### 5.2 Green Method for Contexts

To use the Green method to find Wikipedia articles which correspond to a given word to be disambiguated, articles which may discuss that word and the context surrounding that word are found in Wikipedia as an initial set of locations for the random walker to start. This is done by looking for the word itself as the name of an article. If there is not an article whose name corresponds to the word in question, then articles with the word as a substring of the article name are found.

Since the goal of WSD is to choose the best word sense within the context of other words, we use a given word’s context to select a set of Wikipedia articles which may discuss the content of the word in question. The expectation is that the context words will aid in disambiguation and that the context words will together be associated with an appropriate sense of the word being disambiguated. For this method we defined a word’s context as the word itself, the content words in the sentence the word occurs in, and those occurring in the sentences before and after that sentence.

### 5.3 Green Method for Senses

Every sense of a word to be disambiguated also needs to be represented as corresponding articles in Wikipedia before using the Green method. The words that we search for in the titles of Wikipedia articles include the word itself, and, for every sense, the content words of the sense’s WordNet gloss, as well as the content of the sense’s hypernym gloss

and the synonyms of the hypernym. Exploring this particular aspect of this module — which information about a sense to extract before using the Green Method — is a point for further exploration.

#### 5.4 Interpreting Projections

The Green method as described by Ollivier and Senellart (2007) uses, as the initial set of articles, the vector containing only one article: that article for which related articles are being searched. We use as the initial set of articles the collection of articles in Wikipedia corresponding to either the context for the word to be disambiguated or the sense of a word. The random walker is modeled as starting in any of the articles in this set with uniform probability. Within the context of the Green method, this means that this initial set of articles corresponds to what would be linked to from a *new* Wikipedia article about the sense or context. Each of the content words in this new article (which is not in Wikipedia) would link to one of the articles in the set found by the methods described above. In this way the results of the Green method computation can be interpreted as a relatedness metric for the sense or context itself and the articles which are in Wikipedia.

#### 5.5 Analysis

The process of finding the sense of a word to be disambiguated is as follows: the vector output from the Green method (a relatedness measure between the initial seed and each article in Wikipedia) for the context of the word is compared against the vector output from using the Green method on each sense that the word could have. The comparison is done using the cosine of the angle between the two vectors.

To determine for which instances in SENSEVAL this method may perform well, an analysis was performed on a small development set (15 sentences) from SemCor. A simple heuristic was formulated, selecting the sense with the nearest Green method output to the sentence’s Green method output when the ratio between the first and second highest ranked senses’ cosine angle scores was above a threshold. Applying this heuristic to the EAW task yielded an expectedly low recall of 11% but a precision of 81% on all the words that this heuristic could apply, but only a precision of 25% (recall 0.5%) for non-monosemous words (which were the desired targets

	MFS	Rerank	Wiki
MFS	–	94%	97%
Rerank	23%	–	99%
Wiki	45%	98%	–

Table 5: Complementarity between modules

of the method). Of 37 instances where this method differs from the MFS baseline in the EAW task, 8 instances are correctly disambiguated by this module.

## 6 Results

Although the individual systems have fairly low recall, we can calculate pairwise complementarity between systems  $s_i$  and  $s_j$  by evaluating

$$\left(1 - \frac{|\text{wrong in } s_i \text{ and } s_j|}{|\text{wrong in } s_i|}\right)$$

The results, presented in Table 5, indicate that the systems complement each other well, and suggest that a combination system could have a higher performance than the individual systems.

We investigate a number of techniques to combine the results – while the integration of the lemma / part of speech refinement is done by all modules as a pre-processing step, the method of combination of the resulting modules is less clear. As shown in Florian et al. (2002), a simple voting mechanism achieves comparable performance to a stacking mechanism. We present our results in Table 6, DT gives the result of a 10-fold cross-validation of WEKA stacked decision trees and nearest neighbours built from the individual system results (Witten and Frank, 2000).

Very few decisions are changed with the voting method of combination, and the overall result does not outperform the best MFS baseline (presented in the table as “All MFS”). This combination method may be more useful with a greater number of systems being combined – our system only combines three systems (thus only one non-MFS system has to suggest the MFS for this to be selected), and backs off to the MFS sense in case all three disagree. The degree of complementarity between the Wiki system and the MFS system indicates that these will override the Rerank system in many cases.

Better results are seen with the simple stacking result: in this case, systems are ordered and thus

System	Precision	Recall	F-measure
All MFS	58.6%	58.6%	58.6%
Voting	58.6%	58.6%	58.6%
Stacking	58.9%	58.9%	58.9%
Stacked DT/NN	58.7%	58.7%	58.7%

Table 6: Resulting refined system (forced-choice)

are not being subjected to overriding by other MFS skewed systems.

## 7 Conclusion

We have presented a refinement of the most frequent sense baseline system, which incorporates a number of novel approaches to word sense disambiguation methods. We demonstrate the need for accurate lemmatization and part of speech tagging, showing that that is probably the area where the biggest boost in performance can currently be obtained. We would also argue that examining the absolute performance in a task where the baseline is so exceedingly variable (ourselves, we have found the baseline to be as low as 56% with restricted lemma backoff, 58.4% with a fairly sophisticated lemma / PoS module, against published baselines of 61.5% in McCarthy et al., 62.5% reported in Snyder, or the upper bound baseline of 66% using correct lemmas and parts of speech), the performance difference between the baseline used and the resulting system is interesting in itself.

## Acknowledgments

We would like to thank DJ Hovermale for his input throughout this project.

## References

Agirre, E., , and de Lacalle Lekuona, O. L. (2004). Publicly Available Topic Signatures for all WordNet Nominal Senses. In *Proceedings of the 4<sup>th</sup> International Conference on Languages Resources and Evaluations (LREC)*, Lisbon, Portugal.

Baayen, H., Piepenbrock, R., and Gulikers, L. (1995). The CELEX lexical database (release 2). CD-ROM. Centre for Lexical Information, Max Planck Institute for Psycholinguistics, Nijmegen;

Linguistic Data Consortium, University of Pennsylvania.

- Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117.
- Briscoe, E., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia.
- Elworthy, D. (1994). Does Baum-Welch re-estimation help taggers? In *Proceedings of the 4th ACL Conference on Applied NLP*, pages 53–58, Stuttgart, Germany.
- Florian, R., Cucerzan, S., Schafer, C., and Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. *Journal of Natural Language Engineering*, 8(4):327–342.
- Halácsy, P., Kornai, A., and Oravecz, C. (2007). HunPos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212, Prague, Czech Republic. Association for Computational Linguistics.
- Korhonen, A., Krymolovski, Y., and Briscoe, T. (2006). A large subcategorization lexicon for natural language processing applications. In *Proceedings of the 5th international conference on Language Resources and Evaluation*, pages 1015–1020.
- Korhonen, A. and Krymolowski, Y. (2002). On the robustness of entropy-based similarity measures in evaluation of subcategorization acquisition systems. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 91–97.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the COLING-ACL’98*, pages 768–773.
- Madani, O. and Connor, M. (2008). Large-Scale

- Many-Class Learning. In *Proceedings of the SIAM Conference on Data Mining (SDM-08)*.
- McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2007). Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.
- Mihalcea, R. (2007). Using Wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York.
- Miller, G., Beckwith, R., Felbaum, C., Gross, D., and Miller, K. (1990). Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.
- Miller, G., Leacock, C., Ranea, T., and Bunker, R. (1993). A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 232–235.
- Ollivier, Y. and Senellart, P. (2007). Finding related pages using Green measures: An illustration with Wikipedia. In *Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI 2007)*.
- Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 38–41.
- Schmidt, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.
- Snyder, B. and Palmer, M. (2004). The english all-words task. In Mihalcea, R. and Chklovski, T., editors, *Proceedings of SENSEVAL-3: Third International Workshop on Evaluating Word Sense Disambiguating Systems*, pages 41–43.
- Witten, I. H. and Frank, E. (2000). *Data mining: Practical Machine Learning Tools and Techniques with Java Implementations*, chapter 8. Morgan Kaufmann Publishers.
- XTAG Research Group (2001). A lexicalized tree adjoining grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania.
- Yarowsky, D. (1993). One Sense Per Collocation. In *Proceedings of the Human Language Technology Conference*, Princeton, NJ, USA.
- Yngve, V. H. (1955). Syntax and the problem of multiple meaning. In Locke, W. N. and Booth, A. D., editors, *Machine translation of languages*, pages 208–226. John Wiley and Sons, New York.