# Combining Heterogeneous Classifiers for Word-Sense Disambiguation

**H. Tolga Ilhan**, **Sepandar D. Kamvar**, **Dan Klein**,
**Christopher D. Manning** and **Kristina Toutanova**
Computer Science Department
Stanford University
Stanford, CA 94305-9040, USA

## Abstract

The Stanford-CS224N system is an ensemble of simple classifiers. The first-tier systems are heterogeneous, consisting primarily of naive-Bayes variants, but also including vector space, memory-based, and other classifier types. These simple classifiers are combined by a second-tier classifier, which variously uses majority voting, weighted voting, or a maximum entropy model. Results from SENSEVAL-2 lexical sample tasks indicate that, while the individual classifiers perform at a level comparable to middle-scoring team's systems, the combination achieves high performance. In this paper, we discuss both our system and lessons learned from its behavior.

## 1 Introduction

The problem of supervised word sense disambiguation (WSD) has been approached using many different classification algorithms, including naive Bayes, decision trees, decision lists, and memory-based learners. While it is unquestionable that certain algorithms are better suited to the WSD problem than others (for a comparison, see Mooney (1996)), it seems to be the case that, given similar features as input, various algorithms do not behave dramatically differently. This was seen in the SENSEVAL-2 results where a large fraction of the systems had scores clustered in a fairly narrow region.

We began building our system with 23 supervised WSD systems, each submitted by a student taking the natural language processing course (CS224N) at Stanford University. Students were free to implement whatever WSD method they chose. While most implemented variants of naive Bayes, some implemented a range of other methods, including $n$-gram models, vector space models, and even memory-based learners. Although none of these systems alone would have produced more than middle-level performance on the SENSEVAL-2 task, we decided to investigate how they would behave in combination.

In section 2, we discuss the first-tier classifiers in greater depth and describe our methods of combination. Section 3 discusses performance, analyzing what benefit was found from combination, and when. We also discuss aspects of the component systems which substantially influenced overall performance.

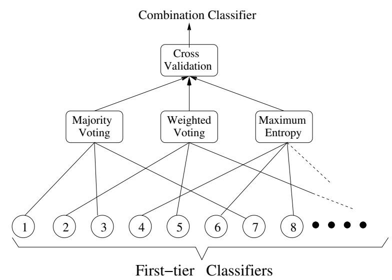


Figure 1: Organization of the system.

## 2 The System

Figure 1 shows the high-level organization of our system. First, each of the 23 classifiers is run with 5-fold cross-validation on the training data. Classifiers are ranked, for each word, based on their held-out accuracy. In any given run of the system, for some $k$, the top $k$ classifiers are kept, while lower-ranking classifiers are discarded. These remaining classifiers are combined by one of three methods.

- *Majority voting:* The sense output by the most classifiers is chosen. Ties are broken in favor of the highest-ranked classifier.

This paper is based on work supported in part by the National Science Foundation under Grants IIS-0085896 and IIS-9982226, by an NSF Graduate Fellowship, and by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University.

- *Weighted voting:* Each classifier is assigned a voting weight (see below) and adds that weight to the sense it outputs. The sense receiving the greatest total weight is chosen.
- *Maximum entropy:* A maximum entropy classifier is trained (see below) and run on the (classifier, vote) outputs from the first tier.

We consider $k$ in the range $\{5, 7, 9, 11, 13, 15\}$, and so, once the ranking of the first-tier classifiers is set, there are 18 possible second-tier classifiers.

We train and test each ($k$, method) pair on the training data, again with 5-fold cross-validation. The classifier type and $k$-value which perform best on the held-out data are chosen. Once the ($k$, method) pair is chosen, all first-tier classifiers, as well as the parameters for the second-tier combinator, are retrained on the entire training corpus. Each target word is considered an entirely separate task, and different first- and second-tier choices can be, and are, made for each word. Table 1 shows what second-tier choices were made for each word.

## 2.1 Combination Methods

Our second-tier classifier takes training instances of the form $\bar{s} = (s, s_1, \ldots, s_k)$ where $s$ is the correct sense and each $s_i$ is the sense chosen by classifier $i$. We initially planned to combine students' classifiers using only a maximum entropy model. Such a model has a set of features $f_x(\bar{s})$ where each feature $f_x$ is true over a subset of vectors $\bar{s}$. A conditional maximum entropy model with such features assigns, for any given choices $s_i$, a distribution over the possible senses $s$. This distribution is of the form:

$$P(s|s_1, \ldots, s_k) = \frac{\exp \sum_x \lambda_x f_x(s, s_1, \ldots, s_k)}{\sum_t \exp \sum_x \lambda_x f_x(t, s_1, \ldots, s_k)}$$

The intent was to design the features to recognize and exploit "sense expertise" in the individual classifiers. For example, one classifier might be trustworthy when reporting a certain sense but less so for other senses. However, there was nowhere near enough data to accurately estimate parameters for such models.[1]

In fact, we noticed that, for certain words, simple majority voting performed better than the maximum entropy model. It also turned out that the most complex features we could get value from were features of the form:

$$f_i(s, s_1, \ldots, s_k) = 1 \iff s = s_i$$

However, with only these features, the maximum entropy approach reduces to a weighted vote; the $s$ which maximizes the posterior probability $P(s|s_1, \ldots, s_k)$ also maximizes the vote:

$$v(s) = \textstyle\sum_i \lambda_i \delta(s_i = s)$$

The indicators $\delta$ are true for exactly one sense, and correspond to the simple $f_i$ defined above.[2] The sense with the highest vote value of $v(s)$ will be the sense with the highest posterior probability $P(s|s_1, \ldots s_k)$ and will be chosen.

All three of our combination schemes can be seen as ways of estimating the weights $\lambda_i$. For majority voting, we skip any attempt at statistical estimation and simply assign each $\lambda_i$ to be $1/k$. For the maximum entropy classifier, we estimate the weights by maximizing the likelihood of a held-out set, using the standard IIS algorithm (Berger et al., 1996).

In weighted voting, we do something in between. We treat the $\delta$ functions as probabilities, treat $v(s)$ as a mixture model, and do a single round of EM to update the $\lambda_i$ starting from uniform weights. As we move from majority voting to weighted voting to maximum entropy, the estimation becomes more sophisticated, but also more prone to overfitting. Since solving overfitting is hard, while choosing between classifiers based on held-out data is relatively easy, this spectrum gives us a way to gracefully handle the range of sparsities in the training corpora for different words.

## 2.2 Individual Classifiers

While our first-tier classifiers implemented a variety of classification algorithms, the differences in their individual accuracies did not primarily stem from the algorithm chosen. Rather, implementation details led to the largest differences. Naive-Bayes classifiers which chose sensible window sizes, or dynamically chose between window sizes tended to outperform those which chose poor sizes. Generally, the optimal windows were either of size one (which

[1] The number of features was not large, only one for each (classifier, chosen sense, correct sense) triple. However, most senses are rarely chosen and rarely correct, and so most features had zero or singleton support.

[2] If the $n$th classifier $e_n$ returns $s$ as the sense, then $\delta(s_n = s)$ is 1, otherwise it is zero.

detected syntactic or collocational cues) or of very large size (which detected more topical cues). Programs with hard-wired window sizes of, say, 5, performed poorly. Ironically, such middle-size windows were commonly chosen by students, but never useful; either extreme was a better design.

Another implementation choice dramatically affecting performance, also for naive-Bayes, was the amount and type of smoothing. Heavy smoothing and smoothing which backed off conditional distributions to the relevant marginal distributions gave good results, while insufficient smoothing or backing off to uniform marginals gave substantially degraded results.[3]

There is one significant way in which our first-tier classifiers were likely different from other teams' systems. In the original class project, students were guaranteed that the ambiguous word would only appear in a single orthographic form. Since this was not true of the SENSEVAL-2 data, we mapped the ambiguous words (but not their context words) down to a citation form. We suspect that this lost quite a bit of information, since there is considerable correlation between form and sense, especially for verbs, but we made no attempt to re-engineer the student systems, and have not thoroughly investigated how big a difference this stemming made.

## 3 Results and Discussion

Table 1 shows the results per word, and table 2 shows results by part-of-speech. A wide range of models are chosen, and the chosen model usually beats the best single classifier for that word, on average by 1.9%. The improvement over the globally best single classifier is even greater.

Notably, if we use the test data as an oracle to chose the best combination method, rather than relying on held-out data, accuracy jumps by an average of 3.6%. This gap is dramatically larger than the gap between the top scoring systems for this SENSEVAL-2 task. While the knowledge of actual best performance is obviously not available, one might suspect that a more sophisticated or better-tuned method of choosing a final combination model might lead to significant improvement.

[3]In particular, there is a defective behavior with naive Bayes where, when one smoothes far too little, the chosen sense is the one which has occurred with the most words in the context window. For skewed-prior data like the SENSEVAL-2 sets, this is invariably the common sense, regardless of what the context words are.

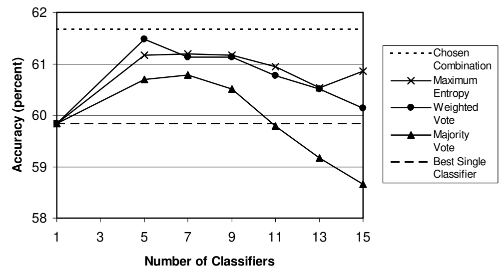


Figure 2: The accuracy of the various combination methods as the number of component systems changes. The *best single classifier* is chosen per word from held-out data and averaged. *Chosen combination* is also selected per word and averaged.

Figure 2 shows how the three combination methods' average scores varied with the number of component classifiers used. A critical aspect of our system is that the first-tier classifiers are very diverse, not only in implementation but also in performance. Initially, accuracy increases as added classifiers bring value to the ensemble. However, as lower-quality classifiers are added in, the better classifiers are steadily drowned out. The weighted vote and maximum entropy combinations are less affected by low-quality classifiers than the majority vote, being able to suppress them with low weights. Still, majority vote was a good method to have around for words where weights could not be usefully set by the other methods.

When combining heterogeneous classifiers, one would like to know when and how the combination will outperform the individuals. One factor is how complementary the mistakes of the individual classifiers are. We can measure this complementarity by averaging, over all pairs of classifiers, the fraction of errors that pair has in common. This gives average pairwise error independence. Another factor is the difficulty of the word being disambiguated. A high most-frequent sense baseline means that there is little room for improvement by combining classifiers. Figure 3 shows, for the overall top 7 first-tier classifiers, the absolute gain between their average accuracy and the accuracy of their majority. The x-axis is the difference between the pairwise independence and the baseline accuracy. The pattern is loose, but clear. The gain increases with complementarity and decreases with the baseline.

| | Single | | Combination | | | Oracle | | Chosen | |
|---|---|---|---|---|---|---|---|---|---|
| word | base | sngl | vot7 | wei7 | me7 | best | any | used | model |
| art-n | 41.8 | 58.2 | 53.1 | 54.1 | 52.0 | 58.2 | 74.5 | 58.2 | wei5 |
| authority-n | 33.7 | 70.7 | 70.7 | 70.7 | 68.5 | 76.1 | 92.4 | 72.8 | wei5 |
| bar-n | 39.7 | 72.2 | 61.6 | 64.9 | 70.2 | 71.5 | 86.8 | 65.6 | me9 |
| begin-v | 58.6 | 81.4 | 82.1 | 82.1 | 86.1 | 86.1 | 95.0 | 84.3 | me15 |
| blind-a | 83.6 | 76.4 | 87.3 | 87.3 | 81.8 | 87.3 | 94.5 | 87.3 | wei7 |
| bum-n | 75.6 | 55.6 | 75.6 | 75.6 | 71.1 | 75.6 | 91.1 | 64.4 | me15 |
| call-v | 25.8 | 25.8 | 31.8 | 30.3 | 24.2 | 33.3 | 65.2 | 25.8 | me5 |
| carry-v | 22.7 | 24.2 | 37.9 | 36.4 | 33.3 | 37.9 | 72.7 | 21.2 | me15 |
| chair-n | 79.7 | 82.6 | 81.2 | 81.2 | 82.6 | 82.6 | 84.1 | 82.6 | me5 |
| channel-n | 27.4 | 60.3 | 58.9 | 60.3 | 63.0 | 67.1 | 86.3 | 60.3 | wei7 |
| child-n | 54.7 | 79.7 | 54.7 | 54.7 | 78.1 | 78.1 | 89.1 | 75.0 | me15 |
| church-n | 53.1 | 73.4 | 75.0 | 75.0 | 75.0 | 76.6 | 90.6 | 75.0 | me5 |
| circuit-n | 27.1 | 78.8 | 64.7 | 64.7 | 72.9 | 78.8 | 89.4 | 78.8 | me5 |
| collaborate-v | 90.0 | 90.0 | 90.0 | 90.0 | 90.0 | 90.0 | 90.0 | 90.0 | wei15 |
| colorless-a | 65.7 | 62.9 | 62.9 | 65.7 | 65.7 | 68.6 | 85.7 | 62.9 | vot7 |
| cool-a | 46.2 | 53.8 | 55.8 | 55.8 | 48.1 | 59.6 | 84.6 | 48.1 | me5 |
| day-n | 59.3 | 62.1 | 68.3 | 69.0 | 64.8 | 69.0 | 84.8 | 67.6 | me5 |
| detention-n | 65.6 | 84.4 | 84.4 | 84.4 | 84.4 | 84.4 | 90.6 | 84.4 | wei5 |
| develop-v | 29.0 | 29.0 | 34.8 | 34.8 | 34.8 | 42.0 | 69.6 | 33.3 | vot13 |
| draw-v | 9.8 | 24.4 | 31.7 | 24.4 | 24.4 | 31.7 | 43.9 | 24.4 | me5 |
| dress-v | 42.4 | 49.2 | 47.5 | 49.2 | 42.4 | 49.2 | 72.9 | 49.2 | wei9 |
| drift-v | 25.0 | 28.1 | 25.0 | 25.0 | 28.1 | 34.4 | 75.0 | 25.0 | vot7 |
| drive-v | 28.6 | 26.2 | 38.1 | 38.1 | 31.0 | 45.2 | 69.0 | 45.2 | wei15 |
| dyke-n | 89.3 | 92.9 | 92.9 | 92.9 | 92.9 | 92.9 | 96.4 | 92.9 | vot5 |
| face-v | 83.9 | 67.7 | 83.9 | 83.9 | 86.0 | 86.0 | 88.2 | 83.9 | wei15 |
| facility-n | 48.3 | 67.2 | 67.2 | 69.0 | 63.8 | 74.1 | 91.4 | 65.5 | wei15 |
| faithful-a | 78.3 | 78.3 | 78.3 | 78.3 | 78.3 | 78.3 | 100 | 78.3 | wei15 |
| fatigue-n | 76.7 | 90.7 | 90.7 | 90.7 | 93.0 | 93.0 | 93.0 | 90.7 | wei7 |
| feeling-n | 56.9 | 49.0 | 56.9 | 56.9 | 60.8 | 60.8 | 88.2 | 56.9 | wei9 |
| find-v | 14.7 | 29.4 | 30.9 | 30.9 | 23.5 | 30.9 | 55.9 | 29.4 | vot13 |
| fine-a | 38.6 | 51.4 | 57.1 | 58.6 | 60.0 | 61.4 | 80.0 | 55.7 | me5 |
| fit-a | 51.7 | 82.8 | 89.7 | 89.7 | 79.3 | 89.7 | 96.6 | 89.7 | wei9 |
| free-a | 39.0 | 53.7 | 57.3 | 57.3 | 61.0 | 61.0 | 75.6 | 61.0 | me9 |
| graceful-a | 75.9 | 79.3 | 79.3 | 79.3 | 79.3 | 79.3 | 89.7 | 79.3 | vot9 |
| green-a | 78.7 | 83.0 | 83.0 | 83.0 | 85.1 | 85.1 | 92.6 | 84.0 | me15 |
| grip-n | 54.9 | 74.5 | 66.7 | 66.7 | 56.9 | 70.6 | 84.3 | 66.7 | me11 |
| hearth-n | 75.0 | 62.5 | 75.0 | 62.5 | 62.5 | 75.0 | 87.5 | 75.0 | vot15 |
| holiday-n | 83.9 | 83.9 | 83.9 | 83.9 | 83.9 | 83.9 | 96.8 | 83.9 | me15 |
| keep-v | 37.3 | 47.8 | 38.8 | 50.7 | 47.8 | 52.2 | 68.7 | 47.8 | me5 |
| lady-n | 69.8 | 77.4 | 79.2 | 79.2 | 77.4 | 79.2 | 83.0 | 79.2 | wei7 |
| leave-v | 31.8 | 40.9 | 42.4 | 45.5 | 37.9 | 45.5 | 75.8 | 43.9 | vot15 |
| live-v | 50.7 | 62.7 | 58.2 | 61.2 | 62.7 | 67.2 | 79.1 | 58.2 | me15 |
| local-a | 57.9 | 68.4 | 71.1 | 71.1 | 68.4 | 73.7 | 92.1 | 68.4 | vot15 |
| match-v | 35.7 | 47.6 | 45.2 | 45.2 | 45.2 | 54.8 | 83.3 | 42.9 | me15 |
| material-n | 42.0 | 46.4 | 53.6 | 53.6 | 50.7 | 60.9 | 88.4 | 58.0 | wei11 |
| mouth-n | 45.0 | 50.0 | 55.0 | 55.0 | 55.0 | 58.3 | 90.0 | 51.7 | vot9 |
| nation-n | 70.3 | 73.0 | 70.3 | 70.3 | 73.0 | 73.0 | 83.8 | 73.0 | me15 |
| natural-a | 27.2 | 55.3 | 47.6 | 47.6 | 47.6 | 55.3 | 79.6 | 52.4 | wei13 |
| nature-n | 45.7 | 45.7 | 45.7 | 45.7 | 56.5 | 58.7 | 84.8 | 45.7 | vot5 |
| oblique-a | 69.0 | 75.9 | 75.9 | 79.3 | 75.9 | 79.3 | 93.1 | 79.3 | wei9 |
| play-v | 19.7 | 37.9 | 39.4 | 40.9 | 37.9 | 45.5 | 68.2 | 40.9 | wei7 |
| post-n | 31.6 | 67.1 | 57.0 | 60.8 | 65.8 | 68.4 | 79.7 | 64.6 | me13 |
| pull-v | 21.7 | 25.0 | 28.3 | 25.0 | 30.0 | 35.0 | 71.7 | 33.3 | me11 |
| replace-v | 53.3 | 53.3 | 53.3 | 53.3 | 53.3 | 55.6 | 88.9 | 53.3 | vot7 |
| restraint-n | 31.1 | 64.4 | 71.1 | 73.3 | 68.9 | 73.3 | 84.4 | 66.7 | wei11 |
| see-v | 31.9 | 37.7 | 43.5 | 43.5 | 39.1 | 43.5 | 60.9 | 40.6 | vot15 |
| sense-n | 22.6 | 52.8 | 60.4 | 58.5 | 52.8 | 64.2 | 83.0 | 60.4 | vot11 |
| serve-v | 29.4 | 54.9 | 60.8 | 62.7 | 58.8 | 66.7 | 76.5 | 56.9 | vot15 |
| simple-a | 51.5 | 54.5 | 51.5 | 51.5 | 54.5 | 54.5 | 83.3 | 53.0 | me5 |
| solemn-a | 96.0 | 96.0 | 96.0 | 96.0 | 96.0 | 96.0 | 96.0 | 96.0 | wei15 |
| spade-n | 63.6 | 63.6 | 78.8 | 78.8 | 81.8 | 81.8 | 81.8 | 75.8 | wei15 |
| stress-n | 46.2 | 48.7 | 35.9 | 41.0 | 51.3 | 51.3 | 89.7 | 51.3 | me9 |
| strike-v | 16.7 | 22.2 | 37.0 | 29.6 | 33.3 | 38.9 | 66.7 | 35.2 | wei15 |
| train-v | 30.2 | 54.0 | 54.0 | 54.0 | 52.4 | 60.3 | 84.1 | 55.6 | wei11 |
| treat-v | 38.6 | 47.7 | 54.5 | 56.8 | 47.7 | 59.1 | 95.5 | 54.5 | vot7 |
| turn-v | 14.9 | 23.9 | 34.3 | 28.4 | 31.3 | 34.3 | 58.2 | 31.3 | wei11 |
| use-v | 65.8 | 64.5 | 65.8 | 65.8 | 65.8 | 68.4 | 81.6 | 65.8 | me9 |
| vital-a | 92.1 | 92.1 | 92.1 | 92.1 | 92.1 | 92.1 | 92.1 | 92.1 | wei15 |
| wander-v | 80.0 | 80.0 | 82.0 | 82.0 | 80.0 | 82.0 | 82.0 | 80.0 | me15 |
| wash-v | 25.0 | 66.7 | 33.3 | 58.3 | 50.0 | 58.3 | 83.3 | 25.0 | vot15 |
| work-v | 26.7 | 50.0 | 45.0 | 41.7 | 43.3 | 45.0 | 76.7 | 41.7 | wei13 |
| yew-n | 78.6 | 78.6 | 78.6 | 78.6 | 78.6 | 78.6 | 82.1 | 78.6 | me15 |

Table 1: Results by word. Single classifiers: *base* = most-frequent-sense baseline, *sngl* = best single first-tier classifier as chosen on held-out data for that word. Fixed combinations: *vot* = majority vote, *wei* = weighted vote, *me* = maximum entropy combination; all are shown for the top seven classifiers only. Oracle bounds: *best* = best combination system as measured on the test data, *any* = test cases where at least one first-tier classifier produced the correct answer. Actually chosen: *model* shows which model performed best according to held-out data, and *used* shows its performance, which were our results for the SENSEVAL-2 English lexical sample task.

| | Single | | Combination | | | Oracle | | Chosen |
|---|---|---|---|---|---|---|---|---|
| | base | sngl | vot7 | wei7 | me7 | best | any | used |
| noun | 50.5 | 67.0 | 65.8 | 66.4 | 67.7 | 71.7 | 86.6 | 68.3 |
| adjective | 57.8 | 67.1 | 68.0 | 68.4 | 67.8 | 71.1 | 86.7 | 68.6 |
| verb | 40.2 | 49.8 | 52.8 | 53.0 | 52.1 | 56.8 | 76.9 | 52.3 |
| average | 47.5 | 59.8 | 60.8 | 61.1 | 61.2 | 65.4 | 82.6 | 61.7 |

Table 2: Results by part-of-speech, and overall.

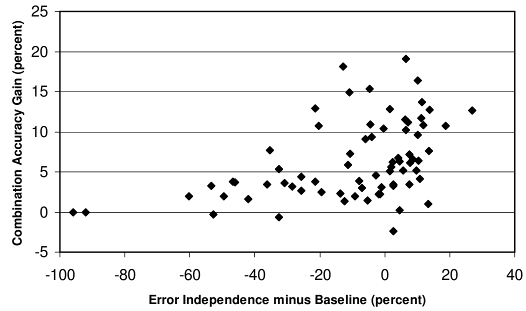


Figure 3: Gain in accuracy of majority vote over the average component performance as (pairwise independence − baseline accuracy) grows.

## 4 Conclusion

We have demonstrated that the combination of a number of heterogeneous classifiers can lead to a substantial performance increase over the individual classifiers. Our system is robust to both the wide range of accuracy of the first-tier classifiers and to sparsity of training data when building the second-tier classifier. The system's overall accuracy is high, despite the medium level of accuracy of the component systems.

## 5 Acknowledgments

We wish to thank the following people for contributing their classifiers to the Stanford-CS224N system: Zoe Abrams, Jenny Berglund, Dmitri Bobrovnikoff, Chris Callison-Burch, Marcos Chavira, Shipra Dingare, Elizabeth Douglas, Sarah Harris, Ido Milstein, Jyotirmoy Paul, Soumya Raychaudhuri, Paul Ruhlen, Magnus Sandberg, Adil Sherwani, Philip Shilane, Joshua Solomin, Patrick Sutphin, Yuliya Tarnikova, Ben Taskar, Kristina Toutanova, Christopher Unkel, and Vincent Vanhoucke.

## References

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

R. J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *EMNLP 1*, pages 82–91.