Some Experiments with a Naive Bayes WSD System

Deniz Yuret
Koc University
Istanbul, Turkey
dyuret@ku.edu.tr

Abstract

This document describes the architecture of a WSD system that participated in the SENSEVAL-3 English all words evaluation exercise. The system uses two independent statistical models, one based on local collocations and another based on a bag of words around the target. The model with the higher confidence provides the final answer for each instance. Both models use Naive Bayes and supervised training with different feature sets. The experiments using this system indicate that the specific smoothing parameters used for Naive Bayes make a big impact on the performance, smaller context sizes give better accuracy, and that the bag of words model adds little to the performance.

1 Introduction

Word sense disambiguation in free text is still an open problem. The best results of the last Senseval-2 English all words task reached 69% accuracy, which is 6% above the baseline of picking the first WordNet sense in the correct part of speech[1] (Mihalcea, 2002).

The established techniques for WSD involve training a machine learning system on sense tagged data using salient features from the context. The features may be local, such as collocations, short distance neighbors and syntactic relations, or distant, such as all content words within a large section of text (Yarowsky and Florian, 2002). The main problem in both cases seems to be lack of sufficient training data. The test cases tend to contain words and features that were never seen in the training data.

In this paper we will use a fairly standard WSD algorithm to look into some questions of interest: What is the extent of the data sparseness problem? What is the relative contribution of the local vs. long distance features? What is the optimal context size for training and testing? How do we smooth the probabilities when most features have zero count?

The next section describes the WSD system in detail. Section 3 will describe the experiments and their results.

2 Description of the WSD System

The system uses two independent statistical models, one based on local collocations and another based on a bag of words around the target. The model with the higher confidence provides the final answer for each instance. I will start with a brief description of the Naive Bayes classifier which is the basis for both models.

Naive Bayes

Given a word w, candidate senses s_i, and features describing the context $f_1, \ldots, f_k$, the Naive Bayes estimate for the best sense is:

$$\arg\max_{s_i} \Pr(f_1|s_i) \ldots \Pr(f_k|s_i) \Pr(s_i|w)$$

There are two types of terms to be estimated: $\Pr(f|s)$, the feature probability and $\Pr(s|w)$, the prior probability. The maximum likelihood estimates for these terms would be $\Pr(f|s) = \mathrm{nfs}/\mathrm{ns}$, and $\Pr(s|w) = \mathrm{nsw}/\mathrm{nw}$ where nfs is the number of times feature f and sense s have been seen together in the training set, etc. The problem is that for many values of f, nfs will be 0. We will use $\Pr(f|s) = (\mathrm{nfs}+\alpha)/(\mathrm{ns}+N)$ where the exact values of α and N are determined by parameter optimization described in Section 3. As the experiments indicate that the choice of the particular smoothing method and parameters have a large impact on the performance I will try to give a detailed account of smoothing.

[1] Senses in WordNet are generally ordered from most to least frequently used. Frequency of use is determined by the number of times a sense is tagged in the various semantic concordance texts.

An Example

The features and the associated computations in the system are best explained using an example. Consider the following sentence from the all-words task, and the word *companion* as the target to be disambiguated.

> Haney peered doubtfully at his drinking *companion* through bleary, tear-filled eyes.

Part of Speech Tagging

The system uses the parts of speech given in the associated Treebank file distributed with the test document and only considers senses in the proper part of speech. In our example, the word *companion* has three noun senses and one verb sense in WordNet 1.7.1, and only the noun senses will be considered.

The Prior Probability

The Naive Bayes model requires a prior probability $\Pr(s|w)$ for each sense. WordNet 1.7.1 includes sense frequency information in the index.sense file. The frequencies for the three noun senses of *companion* are given 25, 9, and 0. To avoid zero probabilities we add one to these counts and use 26/37, 10/37, 1/37 as our priors.

The Bag of Words Model

The bag-of-words model uses the words within a certain radius of the target word as the features of the test instance. In the final system, all non-skip words within 128 words of the target were considered as features. The skip words were taken from the file stoplist.pl distributed with WordNet 1.6. All words are lowercased but not stemmed. The list of features for our example contains 39 words after the elimination of skips:

> man, haney, peered, doubtfully, drinking, companion, bleary, tear, filled, eyes, ready, answer, ich, ..., burning, noticed, wondered, tied

The model uses WordNet glosses and pointers as training data. For each synset, all non-skip words within its gloss, morphological variants of the lemmas in the synset, and the lemmas of all the neighbor synsets that are linked with a single pointer are considered as the training data. The training data for the correct sense of companion thus consists of:

> a person who is frequently in the company of another; "drinking companions"; "comrades in arms" associates familiars fellows companions comrade companion fellow associate comrades familiar friend friends dates date escorts escort playfellows playmates playmate playfellow

There are 27 non-skip words in the training data. These include only 2 of the 39 features from the test instance: "drinking", and "companion". The others have zero count, thus we need to use smoothing. For the bag-of-words model the parameters used for smoothing are $\alpha = 2$ and $N = 65536$. The Naive Bayes product for this sense is:

$$\Pr(s|w)\Pr(f_1|s)\Pr(f_2|s)\ldots\Pr(f_{39}|s) = \frac{26}{37}\,\frac{(1+2)(1+2)(0+2)\ldots(0+2)}{(27+65536)^{39}}$$

When we normalize this product by dividing it with the sum of the Naive Bayes products of all three senses we get 77.88% as the probability of this sense.

The Local Collocation Model

The features representing local collocations around the target word for our example are:

HEAD=companion
HEAD_through
HEAD_through_bleary
drinking_HEAD
drinking_HEAD_through
drinking_HEAD_through_bleary

The first entry gives the exact form of the head word. The remaining entries include varying amounts of left and right context up to the first non-skip word. WordNet 1.6 stoplist.pl is used to determine skip words.

The training data for the local collocation model consists of the following publicly available sense tagged data: SemCor 1.7.1 (222199), DSO Corpus of Sense-Tagged English (160225), WordNet 1.7.1 glosses (44902), Open Mind Word Expert (29382), Senseval2 lexsample (13290), Senseval3 lexsample (8391). The numbers indicate the number of instances in each source.

In our example, the synset associated with the correct sense was observed 18 times in the training data. "HEAD=companion" was observed in 6 of these and the collocation "drinking_HEAD" was observed once. The other features have zero counts. For the local collocation model the smoothing parameters used are $\alpha = 0.15$ and $N = 5000$. The Naive Bayes product for this sense is

$$\Pr(s|w)\Pr(f_1|s)\Pr(f_2|s)\ldots\Pr(f_6|s) = \frac{26}{37}\frac{(6+0.15)(1+0.15)(0+0.15)\ldots}{(18+5000)^6}$$

When we normalize this product by dividing it with the sum of the Naive Bayes products of all three senses we get 99.04% for the probability of the correct sense.

Tie Breaking

If a model assigns equal probability to more than one sense, the one with the smallest WordNet sense number is preferred.

Merging Results

In our example, both models predicted the correct sense, so this would be the answer given by the program. However had they predicted two different senses, the answer returned by the local-collocation model would be used because its confidence 99.04% is higher than the confidence of the bag-of-words model 77.88%.

3 Experiments

Naive Bayes Smoothing

Naive Bayes has long been a popular WSD tool (Mooney, 1996; Pedersen, 2000) and according to (Yarowsky and Florian, 2002) it is one of the best performing algorithms. However usually little detail is given on smoothing methods used with Naive Bayes.

To compare various smoothing methods, a set of experiments were run with a Naive Bayes algorithm on the SemCor corpus with six fold cross validation. The algorithm used the words in the same sentence as bag-of-words features and tried to disambiguate all tagged words.

The add-one smoothing as described in (Mitchell, 1997) smooths all frequencies r/n with $(r+1)/(n+V)$ where V is the vocabulary size taken here to be 100000. Naive Bayes with add-one smoothing did 0.5% better than the baseline of picking the first WordNet sense.

For low sample sizes Good-Turing is known as a better estimator of frequency (Gale and Church, 1994). However, in the context of Naive Bayes, Good-Turing estimation results in a significant 4.5% performance *loss* compared to the baseline.

Lacking a satisfactory theory to explain these results I decided to use the additive smoothing with both the numerator and the denominator as adjustable parameters to be optimized. Thus a frequency r/n was smoothed as $(r+\alpha)/(n+N)$ where both α and N are adjustable parameters. The parameters used by the final system were selected using a simple search of the parameter space.

Context size for training and testing

Another important set of parameters for the bag of words model is the size of the context window for training and testing. See (Pedersen, 2000) for previous work on window sizes.

A number of experiments were run using the SemCor data for training and the Senseval-2 all words task data for testing. Training and testing window radii from 1 to 64 were tried, also optimizing the N and α parameters for each case. The best results were achieved using small test window sizes. Setting both the training and testing radius to 8 results in 2.66% improvement above the baseline whereas using window sizes of 64 achieves no improvement above the baseline.

Unfortunately all the experiments with SemCor underperformed the model that just uses WordNet glosses for training. In fact, when both WordNet glosses and the SemCor data was used together the performance was worse than using the glosses alone. Various attempts to use semi-supervised methods to enhance SemCor with similar contexts from untagged data failed.

The fact that small test window sizes consistently outperformed larger ones with SemCor indicates the importance of local collocations. The best performing system in Senseval-2 (Mihalcea, 2002) uses almost exclusively local features. As a result, I have decided to use only WordNet glosses and pointers for the bag of words model and implement an independent local collocation model.

Merging models

The final system consists of two independent models, one based on bag of words, another on local collocations. The model with the higher confidence provides the final answer for each instance. It may be useful to find out how different their answers are and whether the combination of models leads to a significant improvement.

The performance of the baseline for the Senseval-3 all words task, picking the first sense, had 60.90% performance (precision and recall are the same because all instances were answered). The final system achieved a performance of 3.23% above the baseline. The bag of words model was 0.59% above the baseline and the local collocation model was 2.65% above the baseline. The union of the correct answers from both models give us 10.29% above the baseline. This is an upper limit that could be achieved if we were able to correctly pick the best model for every instance. As things stand however, the bag of words model does not seem to add much to the performance.

4 Conclusion

The percentage improvement above the baseline is very small. The main reason seems to be data sparseness. You may have noticed that in the *companion* example even though we used a nontrivial probabilistic model, the final decision was largely due to a single word in the context: *drinking*. It would be interesting to see if this is a typical situation.

In the bag of words model 50% of the instances in the Senseval-3 all words task did not contain any context words recognized by the model other than the target word itself. 23% had a single context word recognized other than the head word as in the *companion* example. This is in spite of the fact that the context is defined with a relatively wide 256 word window.

In the local collocation model 40% of the instances match nothing but the HEAD=xxx feature. 36% of the instances match one or two other patterns which usually involve only function words.

These figures show that only in less than a quarter of the instances a decision is made based on a neighboring content word. On all other instances, we fall back on the first WordNet sense. The key to significantly better accuracy lies in finding a way to learn more from each example, possibly utilizing untagged data or semantic resources.

References

W. Gale and K. Church. 1994. What's wrong with adding one? In N. Oostdijk and P. de Haan, editors, *Corpus based research into language*. Rodopi.

Rada F. Mihalcea. 2002. Word sense disambiguation with pattern learning and automatic feature selection. *Natural Language Engineering*, 8(4):343–358.

Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill.

R. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *EMNLP-96*.

Ted Pedersen. 2000. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *NAACL-00*.

David Yarowsky and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310.