An Ensemble Approach to Corpus–Based Word Sense Disambiguation

Ted Pedersen

This paper presents a corpus–based approach to word sense disambiguation that combines a number of Naive Bayesian classifiers into an ensemble that performs disambiguation via a majority vote. Each of the member classifiers is based on collocation and co–occurrence features found in varying sized windows of context. This approach is motivated by the observation that, in general, enhancing the feature set or learning algorithm used by a corpus–based approach does not improve disambiguation accuracy beyond what can be attained with shallow lexical features and the Naive Bayesian classifier. Despite the simplicity of this approach, empirical results disambiguating the widely studied nouns *line* and *interest* show that such an ensemble achieves levels of accuracy comparable to the best previously published results.

1 Introduction

Word sense disambiguation is the process of selecting the most appropriate meaning for a word, based on the context in which it occurs. For example, *bill* has several possible meanings; a piece of currency, pending legislation, a bird jaw, etc. However, when used in the context of *The Senate bill is under consideration*, a human reader immediately understands that *bill* is being used in the legislative sense. Humans posses a lifetime of knowledge about the world and experience with language and rarely need to consciously think about sense disambiguation. However, a computer program attempting to perform the same task faces a difficult problem since it does not have the benefit of innate common–sense or linguistic knowledge.

Rather than attempting to provide computer programs with real–world knowledge comparable to that of humans, natural language processing has turned to corpus–based methods. These approaches use techniques from statistics and machine learning to induce models of language usage from large samples of text. These models are trained to perform particular tasks, usually via supervised learning. In this framework, the learner is presented with a number of examples that demonstrate the correct outcome for a

problem. In word sense disambiguation, these examples consist of a number of sentences where each instance of an ambiguous word has been manually annotated with a label that denotes its proper sense.[1]

A sense–tagged corpus must be converted into a form suitable for a supervised learning algorithm. This requires that each sense–tagged usage of an ambiguous word be represented by a set of features that characterize the context in which the word occurs. Any properties of the ambiguous word and the surrounding context that are relevant to disambiguation should be identified and represented as features. Given the flexibility and complexity of human language, there is potentially an infinite set of features that could be utilized. However, in corpus–based approaches features usually consist of information that can be extracted or inferred fairly directly from the text, without relying on extensive amounts of real–world knowledge. These typically include the part–of–speech of surrounding words, the presence of certain key words within some window of context, and various syntactic properties of the sentence.

This paper continues with an overview of a proposed ensemble methodology for disambiguation. Then the Naive Bayesian classifier is introduced, as are the features used to represent the context in which ambiguous words occur. This is followed by a description of the proposed methodology for formulating the ensemble. Then, the *line* and *interest* data is described. Experimental results disambiguating these words with an ensemble of Naive Bayesian classifiers are shown to rival previously published results. This paper closes with a discussion of the choices made in formulating this methodology.

2 Ensembles for Corpus–Based Disambiguation

This paper presents a corpus–based approach that results in high accuracy by combining a number of very simple classifiers into an ensemble that performs disambiguation via a majority vote. This approach is motivated by the observation that, in general, enhancing the feature set or learning algorithm of a corpus–based approach does not improve disambiguation accuracy beyond what can be obtained with shallow lexical features and a basic supervised learning algorithm.

[1]The process of creating these examples is referred to as sense–tagging and results in a sense–tagged corpus.

For example, rather than learning a representative model of a sense–tagged corpus, a Naive Bayesian classifier [Duda and Hart, 1973] is based on certain blanket assumptions about the interactions among features in a corpus. Despite such assumptions, this proves to be among the most accurate techniques in comparative studies of corpus–based word sense disambiguation methodologies. These studies represent the context in which each instance of a sense-tagged word occurs with a variety of features including part–of–speech and other grammatical information for surrounding words, as well as lexical features describing co-occurrence and collocation of words. However, when the contribution of each type of feature to overall accuracy is analyzed (eg. [Ng and Lee, 1996]), shallow lexical features such as co–occurrence and collocations prove to be stronger contributors to accuracy than do deeper, linguistically motivated features.

It has also been demonstrated in a wide range of domains that the combined accuracy of an ensemble of multiple classifiers is often significantly greater than that of any of the individual classifiers that make up the ensemble (e.g., [Dietterich, 1997]). This observation, combined with the previous history of disambiguation success using shallow lexical features and Naive Bayesian classifiers, suggests that disambiguation accuracy might best be improved by combining the output of a number of simple, yet accurate, classifiers into an ensemble.

2.1 Naive Bayesian Classifiers

In general, corpus–based statistical approaches cast natural language processing tasks as classification problems. The learned probabilistic models indicate the most likely value for a variable that represents the membership category or classification of an event, given the values of other feature variables that represent the context in which that event occurs. In word sense disambiguation the classification variable represents the sense of a particular word and the context in which it occurs is represented by feature variables.

A Naive Bayesian classifier assumes that all the feature variables representing a problem are conditionally independent given the value of a classification variable. In this paper, the context in which an ambiguous word occurs is represented by the feature variables $(F_1, F_2, \ldots, F_n)$ and the sense of the ambiguous word is represented by the classification variable (S). In this paper, all feature variables F_i are binary and represent whether or not a particular word occurs within some number of words to the left or right of an ambiguous word, i.e., a window of context. For a Naive Bayesian clas-

sifier, the joint probability of observing a certain combination of contextual features with a particular sense is expressed as:

$$p(F_1, F_2, \ldots, F_n, S) = p(S) \prod_{i=1}^{n} p(F_i|S) \quad (1)$$

The parameters of this model are $p(S)$ and $p(F_i|S)$. The sufficient statistics, i.e., the summaries of the data needed for parameter estimation, are the frequency counts of the events described by the interdependent variables (F_i, S). In this paper, these counts are the number of sentences in the sense–tagged text where the word represented by F_i occurs within some specified window of context of the ambiguous word when it is used in sense S.

Any parameter that has a value of zero indicates that the associated word never occurs with the specified sense value. These zero values are smoothed by assigning them a very small default probability. Once all the parameters have been estimated, the model has been trained and can be used as a classifier to perform disambiguation by determining the most probable sense for an ambiguous word, given the context in which it occurs.

2.2 Representation of Context

The contextual features used in this paper are binary and indicate if a given word occurs within some number of words to the left or right of the ambiguous word.[2] No additional positional information is contained in these features; they simply indicate if the word occurs within some number of surrounding words.

Punctuation and capitalization are removed from the windows of context. However, there is no stop–list used to remove non–content words. This is a variation on the frequently used *bag–of–words* feature set, where a single window of context includes words that occur to both the left and right of the ambiguous word. In this work there are two windows of context, one representing words that occur to the left of the ambiguous word and another for those to the right. The size of these windows are 0, 1, 2, 3, 4, 5, 10, 25, and 50 words.

[2] Words that occur within 1 or 2 positions of the ambiguous word are considered collocates, while those that occur further away are co–occurrences.

2.3 Ensembles of Naive Bayesian Classifiers

The first step in developing an ensemble is to train a separate Naive Bayesian classifier for each of the 81 possible combination of left and right window sizes:

```
for left_window_size = (0, 1, 2, 3, 4, 5, 10, 25, 50)
    for right_window_size = (0, 1, 2, 3, 4, 5, 10, 25, 50)
        train Naive_Bayes (left_window_size,right_window_size)
    end for
end for
```

Thus, Naive_Bayes (l,r) represents a classifier where the model parameters have been estimated based on frequency counts of shallow lexical features from two windows of context; one including l words to the left of the ambiguous word and the other including r words to the right. Note that Naive_Bayes (0,0) includes no words to the left or right; this classifier acts as a majority classifier that assigns every instance of an ambiguous word to the most frequent sense in the training data. Once the individual classifiers are trained they are evaluated using previously held–out test data.

A key step in creating an ensemble is selecting the classifiers to include as members. The approach here is to group the 81 Naive Bayesian classifiers into general categories representing the sizes of the windows of context. There are three such ranges; *collocate* corresponds to windows 0, 1 and 2 words wide, *mix* to windows 3, 4, and 5 words wide, and *co-occur* to windows 10, 25, and 50 words wide. There are nine possible range categories since there are separate left and right windows. For example, Naive_Bayes(1,3) belongs to the range category (collocate, mix) since it is based on a one word window to the left and a three word window to the right. The most accurate classifier in each of the nine range categories is selected for inclusion in the ensemble. Each of the nine member classifiers votes for the most probable sense given the particular context represented by that classifier; the ensemble disambiguates by assigning the sense that receives a majority of the votes.

3 Experimental Data

The *line* data was created by [Leacock, Towell, and Voorhees, 1993] by tagging every occurrence of *line* in the ACL/DCI Wall Street Journal corpus and the American Printing House for the Blind corpus with one of six possible WordNet senses. These senses and their frequency distribution are shown

sense	frequency
product	2218
written or spoken text	405
telephone connection	429
formation of people or things; queue	349
an artificial division; boundary	376
a thin, flexible object; cord	371
total	4148

Figure 1: Distribution of senses for *line* – the experiments in this paper and previous work use a uniformly distributed subset of this corpus, where each sense occurs 349 times.

in Figure 1. This data has since been used in studies by [Mooney, 1996], [Towell and Voorhees, 1998], and [Leacock, Chodorow, and Miller, 1998]. In previous work and in this paper, a subset of the corpus is utilized such that each sense is uniformly distributed; this reduces the accuracy of the majority classifier to 1/6. The uniform distribution is created by randomly sampling 349 sense–tagged examples from each sense, resulting in a training corpus of 2094 sense–tagged sentences.

The *interest* data was created by [Bruce and Wiebe, 1994] by tagging all occurrences of *interest* in the ACL/DCI Wall Street Journal corpus with senses from the Longman Dictionary of Contemporary English. The experiments in this paper use the entire 2,368 sense–tagged sentence corpus. The senses and their frequency distribution are shown in Figure 2. Unlike *line*, the sense distribution is skewed; the majority sense occurs in 53% of the sentences, while the smallest minority sense occurs in less than 1%.

4 Experimental Results

Eighty-one Naive Bayesian classifiers were trained and tested with the *line* and *interest* data. Ten–fold cross validation was employed; all of the sense–tagged examples for a word were randomly shuffled and divided into ten equal folds. Nine folds were used to train the Naive Bayesian classifier while the remaining fold served as a held–out test set to evaluate the learned classifiers. This process is repeated ten times so that each fold serves as the test set once. The average accuracy of the individual Naive Bayesian

sense	frequency
money paid for the use of money	1252
a share in a company or business	500
readiness to give attention	361
advantage, advancement or favor	178
activity, etc. that one gives attention to	66
quality of causing attention of be given to	11
total	2368

Figure 2: Distribution of senses for *interest* – the experiments in this paper and previous work use the entire corpus, where each sense occurs the number of times shown above.

classifiers across the ten folds is reported in Figures 3 and 4.[3]

Each classifier is based upon a distinct representation of context since each employs a different combination of right and left window sizes. The size and range of the left window of context is indicated along the horizontal margin in Figures 3 and 4 while the right window size and range is shown along the vertical margin. Thus, the boxes that subdivide each figure correspond to a particular range category. The classifier that achieves the highest accuracy in each range category is included as a member of the ensemble; the accuracy these attain are shown in italics. In case of a tie, as in the (mix,mix) range for *interest*, the classifier with the smallest total window of context is included in the ensemble.

The most accurate single classifier for *line* is Naive_Bayes (4,25), which attains accuracy of .836. The accuracy of the ensemble created from the most accurate classifier in each of the range categories is .883, an improvement of nearly 5% in accuracy. The single most accurate classifier for *interest* is Naive_Bayes(4,1), which attains accuracy of .861. The ensemble approach reaches .894, an improvement of just over 3%. The increase in accuracy achieved by both ensembles over the best individual classifier is statistically significant, as judged by McNemar's test with $p = .01$.

[3]The standard deviations were between .01 and .025 and are not shown given their relative consistency.

	50	.628	.734	.798	.820	.834	.828	.828	.825	.834
co–occur	25	.627	.737	.797	.818	*.836*	.831	.825	.833	.833
	10	.620	.745	*.805*	.819	.829	.828	.833	.832	*.835*
	5	.610	.746	.796	.806	.816	.819	.823	.820	*.830*
mix	4	.597	.728	*.798*	.815	.819	.821	.818	.821	.820
	3	.584	.727	.790	.815	.825	*.830*	.817	.814	.817
	2	.534	.714	*.792*	.813	*.820*	.818	*.814*	.810	.811
collocate	1	.416	.680	.779	.790	.798	.793	.800	.806	.809
	0	.137	.583	.734	.766	.786	.788	.788	.791	.801
		0	1	2	3	4	5	10	25	50
		collocate				mix		co–occur		

Figure 3: Accuracy of Naive Bayesian classifiers for *line* - the left window size is on the horizontal and the right window size is on the vertical – the accuracy of an ensemble based on the most accurate classifier from each range category is .883

	50	.735	.803	.819	.828	.826	.825	.815	.802	.806
co–occur	25	.733	.802	.822	.828	.825	.825	.814	.802	.804
	10	.745	.824	*.841*	*.848*	.845	.844	*.823*	.807	.807
	5	.731	.828	.852	.858	.853	.845	.833	.811	.810
mix	4	.717	.834	.850	.851	.843	.844	.829	.806	.804
	3	.699	.835	*.857*	*.858*	.858	.850	*.834*	.810	.804
	2	.658	.832	*.852*	.858	.859	.844	*.826*	.803	.802
collocate	1	.627	.824	.851	.854	*.861*	.847	.822	.807	.801
	0	.529	.721	.773	.779	.785	.766	.769	.760	.754
		0	1	2	3	4	5	10	25	50
		collocate				mix		co–occur		

Figure 4: Accuracy of Naive Bayesian classifiers for *interest* - the left window size is on the horizontal and the right window size is on the vertical – the accuracy of an ensemble based on the most accurate classifier from each range category is .894

	accuracy	method	feature set
Naive Bayesian Ensemble	89%	ensemble of 9	varying left & right b-o-w
Ng & Lee, 1996	87%	nearest neighbor	p-o-s, morph, co-occur collocates, verb-obj
Bruce & Wiebe, 1994	78%	model selection	p-o-s, morph, co–occur
Pedersen & Bruce, 1997	78% 74%	decision tree naive bayes	p-o-s, morph, co–occur

Figure 5: Comparison to previous results for *interest*

	accuracy	method	feature set
Naive Bayesian Ensemble	88%	ensemble of 9	varying left & right b-o-w
Towell & Voorhess, 1998	87%	neural net	local & topical b-o-w, p-o-s
Leacock et. al., 1998	84%	naive bayes	local & topical b-o-w, p-o-s
Leacock et. al., 1993	76% 72% 71%	neural net content vector naive bayes	2 sentence b-o-w
Mooney, 1996	72% 71%	naive bayes perceptron	2 sentence b-o-w

Figure 6: Comparison to previous results for *line*

4.1 Comparison to Previous Results

These experiments use the same sense–tagged corpora for *interest* and *line* as previous studies. A brief summary of the results for *interest* is shown in Figure 5 and for *line* in Figure 6. The accuracy attained by the Naive Bayesian ensemble rivals that of any of the other approaches. However, due to variations in experimental methodologies,[4] it can not be concluded that the differences among the most accurate methods are statistically significant. However, this result is encouraging for Naive Bayesian ensembles since, despite their simplicity, they attain high accuracy that is comparable to the best published results for this data.

4.1.1 Interest

The *interest* data was first studied by [Bruce and Wiebe, 1994]. They employ a representation of context that includes the part–of–speech of the two

[4] For example, in this work 10-fold cross validation is employed to assess accuracy while [Ng and Lee, 1996] train and test using 100 randomly sampled sets of data. Similar differences in training and testing methodology exist among the other studies.

words surrounding *interest*, a morphological feature indicating whether or not *interest* is singular or plural, and the three most statistically significant co–occurring words in the sentence with *interest*, as determined by a test of independence. These features are abbreviated as *p-o-s*, *morph*, and *co–occur* in Figure 5. A decomposable probabilistic model is induced from the sense–tagged corpora using a backward sequential search where candidate models are evaluated with the log–likelihood ratio test. The selected model was used as a probabilistic classifier on a held–out set of test data and achieved accuracy of 78%.

The *interest* data was included in a study by [Ng and Lee, 1996], who represent the context of an ambiguous word with the part–of speech of three words to the left and right of *interest*, a morphological feature indicating if *interest* is singular or plural, an unordered set of frequently occurring keywords that surround *interest*, local collocations that include *interest*, and verb–object syntactic relationships. These features are abbreviated *p-o-s*, *morph*, *co–occur*, *collocates*, and *verb–obj* in Figure 5. A nearest–neighbor classifier was employed and achieved an average accuracy of 87% over repeated trials using randomly drawn training and test sets.

[Pedersen, Bruce, and Wiebe, 1997] and [Pedersen and Bruce, 1997] utilize the original Bruce and Wiebe feature set for the *interest* data. The first compares a range of probabilistic model selection methodologies and finds that none outperform the Naive Bayesian classifier, which attains accuracy of 74%. The second compares a range of machine learning algorithms and finds that a decision tree learner (78%) and a Naive Bayesian classifier (74%) are most accurate.

4.1.2 Line

The *line* data was initially studied by [Leacock, Towell, and Voorhees, 1993]. They evaluate the disambiguation accuracy of a Naive Bayesian classifier, a content vector, and a neural network. The context of an ambiguous word is represented by a bag–of–words where the window of context is two sentences wide. This feature set is abbreviated as *2 sentence b-o-w* in Figure 6. When the Naive Bayesian classifier is evaluated words are not stemmed and capitalization remains. However, with the content vector and the neural network words are stemmed and words from a stop–list are removed. They report no significant differences in accuracy among the three approaches; the Naive Bayesian classifier achieved 71% accuracy, the content vector 72%, and the neural network 76%.

The *line* data was studied again by [Mooney, 1996], where seven different machine learning methodologies are compared. All learning algorithms represent the context of an ambiguous word using the bag–of–words with a two sentence window of context. In these experiments words from a stop–list are removed, capitalization is ignored, and words are stemmed. The two most accurate methods in this study proved to be a Naive Bayesian classifier (72%) and a perceptron (71%).

The *line* data was recently revisited by both [Towell and Voorhees, 1998] and [Leacock, Chodorow, and Miller, 1998]. The former take an ensemble approach where the output from two neural networks is combined; one network is based on a representation of local context while the other represents topical context. The latter utilize a Naive Bayesian classifier. In both cases context is represented by a set of topical and local features. The topical features correspond to the open–class words that occur in a two sentence window of context. The local features occur within a window of context three words to the left and right of the ambiguous word and include co–occurrence features as well as the part–of–speech of words in this window. These features are represented as *local* & *topical* *b-o-w* and *p-o-s* in Figure 6. [Towell and Voorhees, 1998] report accuracy of 87% while [Leacock, Chodorow, and Miller, 1998] report accuracy of 84%.

5 Discussion

The word sense disambiguation ensembles in this paper have the following characteristics:

The members of the ensemble are Naive Bayesian classifiers. In recent years the Naive Bayesian classifier has emerged as a consistently strong performer in a wide range of comparative studies of machine learning methodologies. A recent survey of such results, as well as possible explanations for its success, is presented in [Domingos and Pazzani, 1997]. A similar finding has emerged in word sense disambiguation, where a number of comparative studies have all reported that no method achieves any greater accuracy than the Naive Bayesian classifier.

The context in which an ambiguous word occurs is represented by a combination of co–occurrence and collocational features extracted from varying sized windows of surrounding words. Co–

occurrence and collocational features are recognized as potent sources of disambiguation information and have been widely employed. While many other types of features have also been studied, it isn't clear that they offer substantial advantages over these shallow lexical features for disambiguation. For example, [Ng and Lee, 1996] report that local collocations alone achieve 80% accuracy disambiguating *interest*, while their full set of features result in 89%. Preliminary experiments conducted for this paper, where feature sets included collocates, co–occurrences, part–of–speech and grammatical information for surrounding words, showed that no combination of features resulted in disambiguation accuracy higher than that achieved with collocations and co–occurrences.

Member classifiers are selected for the ensembles based on their performance relative to others with comparable window sizes. The most accurate classifier from each of nine possible category ranges is selected as a member of the ensemble. This is based on preliminary experiments that showed that member classifiers with similar sized windows of context often result in little or no overall improvement in disambiguation accuracy. This was expected since slight differences in window sizes lead to roughly equivalent classifiers that have little opportunity for collective improvement. For example, an ensemble was created for *interest* using the nine classifiers in the range category (medium, medium). The accuracy of this ensemble was .841, slightly less than the most accurate individual classifiers in that range which achieved accuracy of .858.

Early experiments also revealed that an ensemble based on a majority vote of all 81 classifiers performed rather poorly. The accuracy for *interest* was approximately 81% and *line* was disambiguated with slightly less than 80% accuracy. The lesson taken from these results was that an ensemble should consist of classifiers that represent as differently sized windows of context as possible; this reduces the impact of redundant errors made by classifiers that represent very similarly sized windows of context. The ultimate success of an ensemble depends on the ability to select classifiers that make complementary errors. This is discussed in the context of combining part–of–speech taggers in [Brill and Wu, 1998]. They provide a measure for assessing the complementarity of errors between two taggers that could be adapted for use with larger ensembles such as the one discussed here, which has nine disambiguators/members.

A majority vote of the member classifiers determines the outcome of the ensemble. In this paper ensemble disambiguation is based on a simple majority vote of the nine member classifiers. In preliminary experiments a more complex scheme of weighting the votes by the estimated joint probability of the Naive Bayesian classifier was also employed. However, accuracy under a weighted vote was poor. For *interest*, the weighted vote resulted in accuracy of .832 while for *line* it resulted in accuracy of .820. Recall that the simple majority vote resulted in accuracy of .894 for *interest* and .883 for *line*.

6 Conclusions

This paper shows that word sense disambiguation accuracy can be improved by combining a number of simple classifiers into an ensemble. A methodology for formulating an ensemble of Naive Bayesian classifiers is presented, where each member classifier is based on co–occurrence and collocation features extracted from a different sized window of context. This approach was evaluated using the widely studied nouns *line* and *interest*, which are disambiguated with accuracy of 88% and 89%, which rivals the best previously published results.

References

[Brill and Wu, 1998] Classifier combination for improved lexical disambiguation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal.

[Bruce and Wiebe, 1994] Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 139–146.

[Dietterich, 1997] Machine–learning research: Four current directions. *AI magazine*, 18(4):97–136.

[Domingos and Pazzani, 1997] On the optimality of the simple Bayesian classifier under zero–one loss. *Machine Learning*, 29:103–130.

[Duda and Hart, 1973] *Pattern Classification and Scene Analysis*. Wiley, New York, NY.

[Leacock, Chodorow, and Miller, 1998] Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165, March.

[Leacock, Towell, and Voorhees, 1993] Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 260–265, March.

[Mooney, 1996] Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–91, May.

[Ng and Lee, 1996] Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Society for Computational Linguistics*, pages 40–47.

[Pedersen and Bruce, 1997] A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 604–609, Providence, RI, July.

[Pedersen, Bruce, and Wiebe, 1997] Sequential model selection for word sense disambiguation. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 388–395, Washington, DC, April.

[Towell and Voorhees, 1998] Disambiguating highly ambiguous words. *Computational Linguistics*, 24(1):125–146, March.

Ted Pedersen is an Assistant Professor at the University of Minnesota Duluth, Department of Computer Science, 10 University Drive, Duluth, MN 55812, USA. He can be reached at tpederse@d.umn.edu.