CORPUS-BASED STATISTICAL SENSE RESOLUTION

Claudia Leacock,[1] *Geoffrey Towell,*[2] *Ellen Voorhees*[2]

[1]Princeton University, Cognitive Science Laboratory, Princeton, New Jersey 08542
[2]Siemens Corporate Research, Inc., Princeton, New Jersey 08540

ABSTRACT

The three corpus-based statistical sense resolution methods studied here attempt to infer the correct sense of a polysemous word by using knowledge about patterns of word co-occurrences. The techniques were based on Bayesian decision theory, neural networks, and content vectors as used in information retrieval. To understand these methods better, we posed a very specific problem: given a set of contexts, each containing the noun *line* in a known sense, construct a classifier that selects the correct sense of *line* for new contexts. To see how the degree of polysemy affects performance, results from three- and six-sense tasks are compared.

The results demonstrate that each of the techniques is able to distinguish six senses of *line* with an accuracy greater than 70%. Furthermore, the response patterns of the classifiers are, for the most part, statistically indistinguishable from one another. Comparison of the two tasks suggests that the degree of difficulty involved in resolving individual senses is a greater performance factor than the degree of polysemy.

1. INTRODUCTION

The goal of this study is to systematically explore the effects of such variables as the number of senses per word and the number of training examples per sense on corpus-based statistical sense resolution methods. To enable us to study the effects of the number of word senses, we selected the highly polysemous noun *line*, which has 25 senses in WordNet.[1]

Automatic sense resolution systems need to resolve highly polysemous words. As Zipf [2] pointed out in 1945, frequently occurring words tend to be polysemous. The words encountered in a given text will have far greater polysemy than one would assume by simply taking the overall percentage of polysemous words in the language. Even though 86% of the nouns in WordNet have a single sense, the mean number of WordNet senses per word for the one hundred most frequently occurring nouns in the Brown Corpus is 5.15, with only eight words having a single sense.

[1] WordNet is a lexical database developed by George Miller and his colleagues at Princeton University.[1]

2. PREVIOUS WORK

Yarowsky [3] compared the Bayesian statistical method with the published results of other corpus-based statistical models. Although direct comparison was not possible due to the differences in corpora and evaluation criteria, he minimizes these differences by using the same words, with the same definition of sense. He argues, convincingly, that the Bayesian model is as good as or better than the costlier methods.

As a pilot for the present study, a two-sense distinction task for *line* was run using the content vector and neural network classifiers, achieving greater than 90% accuracy. A three-sense distinction task was then run, which is reported in Voorhees, *et. al.* [4], and discussed in Section 5.

3. METHODOLOGY

The training and testing contexts were taken from the 1987-89 *Wall Street Journal* corpus and from the APHB corpus.[2] Sentences containing '[Ll]ine(s)' were extracted and manually assigned a single sense from WordNet. Sentences containing proper names such as 'Japan Air Lines' were removed from the set of sentences. Sentences containing collocations that have a single sense in WordNet, such as *product line* and *line of products*, were also excluded since the collocations are not ambiguous.

Typically, experiments have used a fixed number of words or characters on either side of the target as the context. In this experiment, we used linguistic units – sentences – instead. Since the target word is often used anaphorically to refer back to the previous sentence, we chose to use two-sentence contexts: the sentence containing *line* and the preceding sentence. However, if the sentence containing *line* is the first sentence in the article, then the context consists of one sentence. If the preceding sentence also contains *line* in the same sense, then an additional preceding sentence is added to the context, creating contexts three or more sentences long.

[2] The 25 million word corpus, obtained from the American Printing House for the Blind, is archived at IBM's T.J. Watson Research Center; it consists of stories and articles from books and general circulation magazines.

The average size of the training and testing contexts is 44.5 words.

The sense resolution task used the following six senses of the noun *line*:

1. a *product*: 'a new line of workstations'
2. a *formation* of people or things: 'stand in line'
3. spoken or written *text*: 'a line from Shakespeare'
4. a thin, flexible object; *cord*: 'a nylon line'
5. an abstract *division*: 'a line between good and evil'
6. a telephone *connection*: 'the line went dead'

The classifiers were run three times each on randomly selected training sets. The set of contexts for each sense was randomly permuted, with each permutation corresponding to one *trial*. For each trial, the first 200 contexts of each sense were selected as training contexts. The next 149 contexts were selected as test contexts. The remaining contexts were not used in that trial. The 200 training contexts for each sense were combined to form a final training set (called the 200 training set) of size 1200. The final test set contained the 149 test contexts from each sense, for a total of 894 contexts.

To test the effect that the number of training examples has on classifier performance, smaller training sets were extracted from the 200 training set. The first 50 and 100 contexts for each sense were used to build the new training sets. The same set of 894 test contexts were used with each of the training sets in a given trial. Each of the classifiers used the same training and test contexts within the same trial, but processed the text differently according to the needs of the method.

4. THE CLASSIFIERS

The only information used by the three classifiers is co-occurrence of character strings in the contexts. They use no other cues, such as syntactic tags or word order. Nor do they require any augmentation of the training contexts that is not fully automatic.

4.1. A Bayesian Approach

The Bayesian classifier, developed by Gale, Church and Yarowsky [5], uses Bayes' decision theory for weighting tokens that co-occur with each sense of a polysemous target. Their work is inspired by Mosteller and Wallace [6], who applied Bayes' theorem to the problem of author discrimination. The main component of the model, a *token*, was defined as any character string: a word, number, symbol, punctuation or any combination. The entire token is significant, so inflected forms of a base word (*wait* vs. *waiting*) and mixed case strings (*Bush* vs. *bush*) are distinct tokens. Associated with each token is a set of *saliences*, one for each sense, calculated from the training data. The salience of a token for a given sense is $\Pr(token|sense)/\Pr(token)$. The *weight* of a token for a given sense is the log of its salience.

To select the sense of the target word in a (test) context, the classifier computes the sum of the tokens' *weights* over all tokens in the context for each sense, and selects the sense with the largest sum. In the case of author identification, Mosteller and Wallace built their models using high frequency function words. With sense resolution, the salient tokens include content words, which have much lower frequencies of occurrence. Gale, *et. al.* devised a method for estimating the required probabilities using sparse training data, since the maximum likelihood estimate (MLE) of a probability – the number of times a token appears in a set of contexts divided by the total number of tokens in the set of contexts – is a poor estimate of the true probability. In particular, many tokens in the test contexts do not appear in any training context, or appear only once or twice. In the former case, the MLE is zero, obviously smaller than the true probability; in the latter case, the MLE is much larger than the true probability. Gale, *et. al.* adjust their estimates for new or infrequent words by interpolating between local and global estimates of the probability.

The Bayesian classifier experiments were performed by Kenneth Church of AT&T Bell Laboratories. In these experiments, two-sentence contexts are used in place of a fixed-sized window of ± 50 tokens surrounding the target word that Gale, *et. al.* find optimal,[3] resulting in a smaller amount of context used to estimate the probabilities.

4.2. Content Vectors

The content vector approach to sense resolution is motivated by the vector-space model of information retrieval systems [8], where each *concept* in a corpus defines an axis of the vector space, and a text in the corpus is represented as a point in this space. The concepts in a corpus are usually defined as the set of word stems that appear in the corpus (e.g., the strings *computer(s), computing, computation(al)*, etc. are conflated to the concept *comput*) minus *stopwords*, a set of about 570 very high frequency words that includes function words (e.g., *the, by, you, that, who*, etc.) and content words (e.g., *be, say*, etc.). The similarity between two texts is computed as a function of the vectors representing the two texts.

[3] Whereas current research tends to confirm the hypothesis that humans need a narrow window of ± 2 words for sense resolution [7], Gale, *et. al.* have found much larger window sizes are better for the Bayesian classifier, presumably because so much information (e.g., word order and syntax) is thrown away.

Product			Formation			Text		
Bayesian	Vector	Network	Bayesian	Vector	Network	Bayesian	Vector	Network
Chrysler	comput	comput	night	wait	wait	Biden	speech	familiar
workstations	ibm	sell	checkout	long	long	ad	writ	writ
Digital	produc	minicomput	wait	checkout	stand	Bush	mr	ad
introduced	corp	model	gasoline	park	checkout	opening	bush	rememb
models	sale	introduc	outside	mr	park	famous	ad	deliv
IBM	model	extend	waiting	airport	hour	Dole	speak	fame
Compaq	sell	acquir	food	shop	form	speech	read	speak
sell	introduc	launch	hours	count	short	Dukakis	dukak	funny
agreement	brand	continu	long	peopl	custom	funny	biden	movie
computers	mainframe	quak	driver	canad	shop	speeches	poem	read
Cord			Division			Phone		
Bayesian	Vector	Network	Bayesian	Vector	Network	Bayesian	Vector	Network
fish	fish	hap	blurred	draw	draw	phones	telephon	telephon
fishing	boat	fish	walking	fine	priv	toll	phon	phon
bow	wat	wash	crossed	blur	hug	porn	call	dead
deck	hook	pull	ethics	cross	blur	Bellsouth	access	cheer
sea	wash	boat	narrow	walk	cross	gab	dial	hear
boat	float	rope	fine	narrow	fine	telephone	gab	henderson
water	men	break	class	mr	thin	Bell	bell	minut
clothes	dive	hook	between	tread	funct	billion	servic	call
fastened	cage	exercis	walk	faction	genius	Pacific	toll	bill
ship	rod	cry	draw	thin	narrow	calls	porn	silent

Table 1: The ten most heavily weighted tokens for each sense of *line* for the Bayesian, content vector and neural network classifiers.

For the sense resolution problem, each sense is represented by a single vector constructed from the training contexts for that sense. A vector in the space defined by the training contexts is also constructed for each test context. To select a sense for a test context, the inner product between its vector and each of the sense vectors is computed, and the sense whose inner product is the largest is chosen.

The components of the vectors are weighted to reflect the relative importance of the concepts in the text. The weighting method was designed to favor concepts that occur frequently in exactly one sense. The weight of a concept c is computed as follows:

$$\begin{aligned} \text{Let } n_s &= \text{number of times } c \text{ occurs in sense } s \\ p &= n_s / \textstyle\sum_{\text{senses}} n_s \\ d &= \text{difference between the two largest } n_s \\ &\quad \text{(if difference is 0, } d \text{ is set to 1)} \end{aligned}$$

$$\text{then } w_s = p * \min(n_s, d)$$

For example, if a concept occurs 6 times in the training contexts of sense 1, and zero times in the other five sets of contexts, then its weights in the six vectors are (6, 0, 0, 0, 0, 0). However, a concept that appears 10, 4, 7, 0, 1, and 2 times in the respective senses, has weights of (1.25, .5, .88, 0, .04, .17), reflecting the fact that it is not as good an indicator for any sense. This weighting method is the most effective among several variants that were tried.

We also experimented with keeping all words in the content vectors, but performance degraded, probably because the weighting function does not handle very high frequency words well. This is evident in Table 1, where 'mr' is highly weighted for three different senses.

4.3. Neural Network

The neural network approach [9] casts sense resolution as a supervised learning paradigm. Pairs of [input features, desired response] are presented to a learning program. The program's task is to devise some method for using the input features to partition the training contexts into non-overlapping sets corresponding to the desired responses. This is achieved by adjusting link weights so that the output unit representing the desired response has a larger activation than any other output unit.

Each context is translated into a bit-vector. As with the content vector approach, suffixes are removed to conflate related word forms to a common stem, and *stopwords* and punctuation are removed. Each concept that appears at least twice in the entire training set is assigned to a bit-vector position. The resulting vector has ones in positions corresponding to concepts in the context and zeros otherwise. This procedure creates vectors with more than 4000 positions. The vectors are, however, extremely sparse; on average they contain slightly more than 17 concepts.

Networks are trained until the output of the unit corresponding to the desired response is greater than the output of any other unit for every training example. For testing, the classification determined by the network is given by the unit with the largest output. Weights in a neural network link vector may be either positive or negative, thereby allowing it to accumulate evidence both for and against a sense.

The result of training a network until all examples are classified correctly is that infrequent tokens can acquire disproportionate importance. For example, the context *'Fine,' Henderson said, aimiably* [sic]. *'Can you get him on the line?'* clearly uses *line* in the *phone* sense. However, the only non-stopwords that are infrequent in other senses are 'henderson' and 'aimiably'; and, due to its misspelling, the latter is conflated to 'aim'. The network must raise the weight of 'henderson' so that it is sufficient to give *phone* the largest output. As a result, 'henderson' appears in Table 1, in spite of its infrequency in the training corpus.

To determine a good topology for the network, various network topologies were explored: networks with from 0 to 100 hidden units arranged in a single hidden layer; networks with multiple layers of hidden units; and networks with a single layer of hidden units in which the output units were connected to both the hidden and input units. In all cases, the network configuration with no hidden units was either superior or statistically indistinguishable from the more complex networks. As no network topology was significantly better than one with no hidden units, all data reported here are derived from such networks.

5. RESULTS AND DISCUSSION

All of the classifiers performed best with the largest number (200) of training contexts. The percent correct results reported below are averaged over the three trials with 200 training contexts. The Bayesian classifier averaged 71% correct answers, the content vector classifier averaged 72%, and the neural network classifier averaged 76%. None of these differences are statistically significant due to the limited sample size of three trials.

The results reported below are taken from trial A with 200 training contexts. Confusion matrices of this trial are given in Tables 2 – 4.[4] The diagonals show the number of correct classifications for each sense, and the off-diagonal elements show classification errors. For example, the entry containing 5 in the bottom row of Table 2 means that 5 contexts whose correct sense is the *product* sense were classified as the *phone* sense.

Ten heavily weighted tokens for each sense for each classifier appear in Table 1. The words on the list seem, for the most part, indicative of the target sense. However, there are some consistent differences among the methods. For example, whereas the Bayesian method is sensitive to proper nouns, the neural network appears to have no such preference.

To test the hypothesis that the methods have different response patterns, we performed the χ^2 test for correlated proportions. This test measures how consistently the methods treat individual test contexts by determining whether the classifiers are making the same classification errors in each of the senses. For each sense, the test compares the off-diagonal elements of a matrix whose columns contain the responses of one classifier and the rows show a second classifier's responses in the same test set. This process constructs a square matrix whose diagonal elements contain the number of test contexts on which the two methods agree.

The results of the χ^2 test for a three-sense resolution task (*product*, *formation* and *text*),[5] indicate that the response pattern of the content vector classifier is very significantly different from the patterns of both the Bayesian and neural network classifiers, but the Bayesian response pattern is significantly different from the neural network pattern for only the *product* sense. In the six-sense disambiguation task, the χ^2 results indicate that the Bayesian and neural network classifiers' response patterns are not significantly different for any sense. The neural network and Bayesian classifiers' response patterns are significantly different from the content vector classifier only in the *formation* and *text* senses. Therefore, with the addition of three senses, the classifiers' response patterns appear to be converging.

The pilot two-sense distinction task (between *product* and *formation*) yielded over 90% correct answers. In the three-sense distinction task, the three classifiers had a

[4] The numbers in the confusion matrix in Table 4 are averages over ten runs with randomly initialized networks.

[5] Training and test sets for these senses are identical to those in the six-sense resolution task.

		Correct Sense					
		Product	Formation	Text	Cord	Division	Phone
Classified Sense	Product	120	7	4	2	4	5
	Formation	9	97	19	6	14	11
	Text	5	26	93	6	20	11
	Cord	2	10	11	129	5	10
	Division	8	8	21	5	103	3
	Phone	5	1	1	1	3	109

Table 2: Confusion matrix for Bayesian classifier (columns show the correct sense, rows the selected sense).

		Correct Sense					
		Product	Formation	Text	Cord	Division	Phone
Classified Sense	Product	139	33	32	5	17	14
	Formation	2	88	15	12	8	5
	Text	3	7	71	3	8	6
	Cord	0	7	7	120	2	5
	Division	0	9	12	4	108	0
	Phone	5	5	12	5	6	119

Table 3: Confusion matrix for content vector classifier (columns show the correct sense, rows the selected sense).

		Correct Sense					
		Product	Formation	Text	Cord	Division	Phone
Classified Sense	Product	122	11	4	1	3	6
	Formation	4	90	17	9	8	2
	Text	9	14	83	4	10	7
	Cord	2	11	13	125	3	3
	Division	4	13	16	4	121	1
	Phone	8	10	16	6	4	130

Table 4: Confusion matrix for neural network classifier (columns show the correct sense, rows the selected sense).

mean of 76% correct,[6] yielding a sharp degradation with the addition of a third sense. Therefore, we hypothesized degree of polysemy to be a major factor for performance. We were surprised to find that in the six-sense task, all three classifiers degraded only slightly from the three-sense task, with a mean of 73% correct. Although the addition of three new senses to the task caused consistent degradation, the degradation is relatively slight. Hence, we conclude that some senses are harder to resolve than others, and it appears that overall accuracy is a function of the difficulty of the sense rather than being strictly a function of the number of senses. The hardest sense to learn, for all three classifiers, was *text*, followed by *formation*. To test the validity of this conclusion, further tests need to be run.

If statistical classifiers are to be part of higher-level NLP tasks, characteristics other than overall accuracy are important. Collecting training contexts is by far the most time-consuming part of the entire process. Until training-context acquisition is fully automated, classifiers requiring smaller training sets are preferred. Figure 1 shows that the content vector classifier has a flatter learning curve between 50 and 200 training contexts than the neural network and Bayesian classifiers, suggesting that the latter two require more (or larger) training contexts. Ease and efficiency of use is also a factor. The three classifiers are roughly comparable in this regard, although the neural network classifier is the most expensive to train.

[6] The Bayesian classifier averaged 76% correct answers, the content vector classifier averaged 73%, and the neural networks 79%.

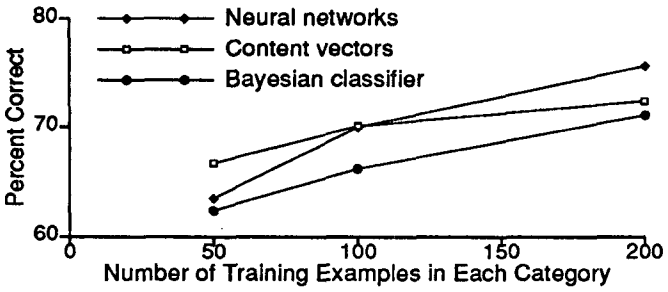

Figure 1: Learning curves.

6. CONCLUSION

The convergence of the response patterns for the three methods suggests that each of the classifiers is extracting as much data as is available in word counts from training contexts. If this is the case, any technique that uses only word counts will not be significantly more accurate than the techniques tested here.

Although the degree of polysemy does affect the difficulty of the sense resolution task, a greater factor of performance is the difficulty of resolving individual senses. Using hindsight, it is obvious that the *text* sense is hard for these statistical methods to learn because one can talk or write about anything. In effect, all words between a pair of quotation marks are noise (unless *line* is within the quotes). In the three-sense task, the Bayesian classifier did best on the *text* sense, perhaps because it had open and closed quotes as important tokens. This advantage was lost in the six-sense task because quotation marks also appear in the contexts of the *phone* sense. It is not immediately obvious why the *formation* sense should be hard. From inspection of the contexts, it appears that the crucial information is close to the word, and context that is more than a few words away is noise.

These corpus-based statistical techniques use an impoverished representation of the training contexts: simple counts of tokens appearing within two sentences. We believe significant increases in resolution accuracy will not be possible unless other information, such as word order or syntactic information, is incorporated into the techniques.

ACKNOWLEDGMENTS

This work was supported in part by Grant No. N00014-91-1634 from the Defense Advanced Research Projects Agency, Information and Technology Office, by the Office of Naval Research, and by the James S. McDonnell Foundation. We thank Kenneth Church of AT&T Bell Laboratories for running the Bayesian classifier experiment, and Slava Katz of IBM's T.J. Watson Research Center for generously supplying *line* contexts from the APHB corpus. We are indebted to George A. Miller for suggesting this line of research.

References

1. Miller, G. A. (ed.), WordNet: An on-line lexical database. *International Journal of Lexicography* (special issue), 3(4):235-312, 1990.
2. Zipf, G. K., The meaning-frequency relationship of words. *Journal of General Psychology*, 3:251-256, 1945.
3. Yarowsky, D., Word-sense disambiguation using statistical models of Roget's categories trained on large corpora, *COLING-92*, 1992.
4. Voorhees, E. M., Leacock C., and Towell, G., Learning context to disambiguate word senses. *Proceedings of the 3rd Computational Learning Theory and Natural Learning Systems Conference, 1992*, MIT Press (to appear). Also available as a Siemens technical report.
5. Gale, W., Church, K. W., and Yarowsky, D., A method for disambiguating word senses in a large corpus. Statistical Research Report 104, AT&T Bell Laboratories, 1992.
6. Mosteller F. and Wallace, D., *Inference and Disputed Authorship: The Federalist.* Addison-Wessley, Reading, MA, 1964.
7. Choueka Y. and Lusignan, S., Disambiguation by short contexts. *Computers and the Humanities*, 19:147-157, 1985.
8. Salton, G., Wong, A., and Yang, C. S., A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613-620, 1975.
9. Rumelhart, D. E., Hinton, G. E., and Williams, R J., Learning internal representations by error propagation. in Rumelhart, D. E. and McClelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.* MIT Press, Cambridge, MA, 1986, pp. 318-363.