A Connectionist Approach to Word Sense Disambiguation

Andy Zuppann
Swarthmore College
CS97: Senior Conference on Natural Language Processing
zuppann@cs.swarthmore.edu

Abstract

Effective word sense disambiguation can play a crucial role in several important computational linguistic tasks. Problems such as information retrieval and machine translation rely upon accurate tagging of word senses. This paper will present an English word sense classifier based upon connectionist models of learning and behavior. Results perform comparably with state of the art statistical approaches in finely grained word sense disambiguation.

1 The Problem of Word Senses

One interesting feature of language, at least from a computational linguistic standpoint, is its inherent ambiguity. Native speakers of a language have very little problem adjusting to potentially ambiguous statements, but both non-native speakers and computers face the difficulty of extracting a specific semantic meaning from statements that could have several.

An archetypical example of this lexical ambiguity is found in the word 'plant.' Given a sentence: *The plant lived in the chemical plant*, a computer attempting, say machine translation, should be aware that each usage of plant in the sentence represents a different sense of the word - in this case, the difference between a living plant and an industrial plant. It is important to correctly identify these difference because the ambiguity is unlikely to be exactly duplicated in the target language. For instance, the French word for the living plant is *plante*, while the word for the factory plant is *usine*. Clearly, a correct translator needs to be able to resolve any sense ambiguity. This paper will describe one such approach for untangling this problem based around neural networks and connectionist models.

2 Previous Work

Standard approachs to this problem have been developed using statistical methods. Various approaches include utilizing assumptions about one sense per discourse and one sense per collocation (Yarowsky, 1993), (Yarowsky, 1995). More recent work challenges and develops these assumptions into complicated statistical models based on topicality and locality of context surrounding a target word to be disambiguated. These models all rely on explicit calculations of the relevance of given context.

One major exercise in disambiguating word senses has been the SENSEVAL project. By preparing corpora in English and several other languages, the program's designers hope to create a forum for comparing the performance of several approaches to the same problem. By specifying exactly the training and testing data for the classifier systems to use, discrepancies between data and results across experiments should be ameliorated and there should be a fair comparison of all the system's disambiguating capabilities. The results of this approach have been promising, and it appears that the state of the art for word sense disambiguation is 75-80% success both in precision and in recall (Kilgarriff, 1998). Furthermore, by making the training and testing corpora used in the exercise widely available, SENSEVAL allows researchers to test and compare new methods against a solid baseline of other systems' performances.

The hypothesis of my work is that, instead of relying on human generated statistical models, a connectionist, developmental approach can yield as good, if not better, results. The foundations of this approach are strongly motivated by a desire to base learning and development in machines on our understanding of our own developmental process and root the learning in biological plausibility. Additionally, studies suggest that this approach can be as successful as other, more traditional approaches to problem solving such as Markov chains and decision trees (Quinlan, 1994).

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College

Although most of the previous work has been focused on resolving sense ambiguity using statistical methods, there still exists substantial evidence that a connectionist approach can lead to comparable results within this particular domain. For instance, Mooney (1996) compares several available word sense classifiers, and out of 7 possible classifiers, a neural net approach tied for best. In this paper, Mooney used a simple perceptron network to disambiguate instances of the word 'line.' The network performs comparably with a Naive Bayesian approach to disambiguation and signficantly better than five other methods, achieving a 70% precision rate. In addition, the neural network approach both trained and tested faster than the Naive Bayesian system.

A separate study also reported a neural net having a high success rate in identifying the meanings of the words 'line,' 'serve,' and 'hard' (Towell and Voorhees, 1998). The study created topical and locational information networks and combined their output to create effective sense classifiers. The topical approach used general context information surrounding a target word. Each word surrounding the ambiguous word in the testing set is given an input into the node, but there is no encoding of any words relation to the target, just that it appears in a similar context.

The locational encoding used by Towell *et al* is a more intricate approach which, when encoding words, affixes locational information. Using their example, the sentence "John serves loyally" becomes the set [-3zzz -2zzz -1John 0serves 1loyally 2. 3zzz]. This affixation massively expands the vocabulary of context words around a target word to contain locational information for each word. Every word within this expanded vocabulary is given its own input node to the network. The locational approach permits a network to uncover for itself not only what context is important, but whether relative location matters as well for disambiguating words. This approach worked extremely well for its three target words, averaging an 86% success rate. This is not altogether surprising, given the rather coarse senses used in their experiment.

My research reported here, to a large degree, is an attempt to reproduce Towell *et al*'s topical neural network model and apply it to a different set of training data. In doing so, I plan to provide two important contributions. One, I will put a neural network model for word sense disambiguation in the context of a previously implemented general word sense exercise comparing different attempts at disambiguation. This will permit accurate comparisons of a neural network to other approachs within a broad framework. Secondly, I hope to test the general connectionist framework for sense tagging in a relatively fine-grained sense environment.

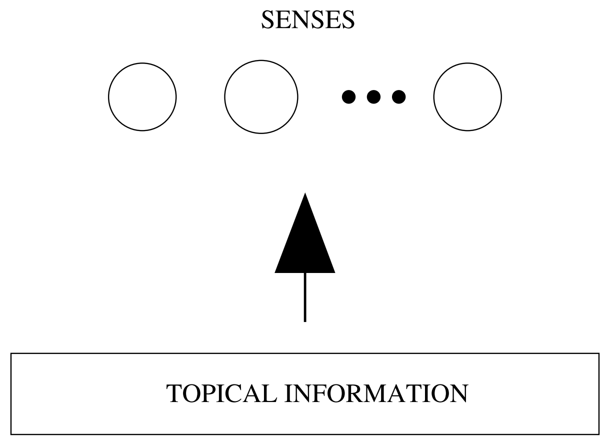

Figure 1: Network Model

3 The Classifier

This section will describe the specifics of my approach. First, the architecture of the network model will be described. Second, the data for training and testing my classifier will be covered. Finally, I will describe the learning method for the network.

3.1 Model

The classifier presented is a very simple neural network comprised of only perceptrons linking topical input to sense output. This idea is based on Towell and Voorhees, who describe a similar system performing - somewhat surprisingly - best without any hidden nodes in the network (Towell and Voorhees, 1998). Indeed, other research into this subject reveals that the poor performance of networks with hidden layers is pervasive (Mooney, 1996)

Given that I will be testing neural networks on their performance on several different potentially ambiguous words, a separate network is required for each. The reason for this is clear when one considers the nature of a single network's inputs for this task. Each network must be able to disambiguate a word's meaning based around that word's particular context and choose out of the word's available senses. This requires the network to have unique inputs and outputs for any target word. To disambiguate a new word, a new network with its own unique parameters must be created and trained.

The general architecture of this model is graphically depicted in Figure 1. A given network will consist of a fixed number of input nodes. The number of input nodes will correspond to the size of the context vocabulary found in the corpus. Any word that appears in a sentence along with the target ambiguous word will have an associated node.

When confronted with an ambiguous word, the net-

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College

work will collect all the words in the surrounding sentence and create an associated vector. This vector will have the length of the vocabulary, and the topical words will have their associated nodes set to 1.0. Other nodes, i.e., words that do not occur in topical context of the current target, will have their activation set to 0.0.

The output of the network will simply be one node per available sense of the word. The node with the highest activation after the network has analyzed the topical input should correspond to a given sense. This sense, then, is the network's classification of the presented instance of the target word.

One important feature of the network is that its structure almost necessitates that recall on tests will be 100%. Although a perfect word sense disambiguator would certainly have recall that high, current efforts have a much lower recall(Kilgarriff and Rosenzweig, 2000). In my network, the precision-recall trade-off can be approximated by setting a threshold of certainty on the output nodes. In other words, during testing, the network only reports results if the highest activated node is greater than all other nodes by a certain margin. Clearly, if two output nodes have similar activation then the system is having a difficult time choosing between the two senses and precision could be improved by not having to tag that instance.

3.2 Data

The data used for training and testing my model comes directly from the SENSEVAL exercise - now referred to as SENSEVAL-1 (Kilgarriff, 1998), (Kilgarriff and Rosenzweig, 2000). The exercise was intended to compare the effectiveness of several different word sense disambiguators on similar data. For the first SENSEVAL, words were seperated lexically before being disambiguated, an approach that fits nicely with my necessity of having one network model per word. The dictionary and corpus for the exercise come from a project called HECTOR, which involved the creation of a comprehensive hand-tagged sense corpus concurrently with the development of a robust dictionary of word senses.

Although SENSEVAL included several different words to disambiguate, I focus on only five of them. Their selection was based primarily on the relative abundance of both training and testing data. My model attempts to disambiguate *accident*, *band*, *brilliant*, *sanction*, and *slight*. Although the HECTOR sense tags are too fine to recreate here in great detail, Table 1 presents a few examples of ambiguities in the target words. A more complete analysis would undoubtedly test on the entire set of SENSEVAL words. It is unfortunate that, given the large training time for a network, I was unable to test my system on the entirety of the SENSEVAL data.

For each word, the SENSEVAL exercise provided training data, a dictionary, testing data, and a gold standard for answers. My system uses all of these directly from the experiment. The resulting network inputs for the training data corresponds to a varied vocabulary range, from a little over 1000 for *sanction* to over 8000 for *band*.

word	*example meanings*
accident	by chance
	collision
band	musical group
	ring
brilliant	showy
	vivid
sanction	allow
	economic penalty
slight	tiny
	least (superlative)

Table 1: Ambiguities in Target Words

word	*Vocabulary Size*	*# of senses*
accident	6129	11
band	8111	22
brilliant	3783	11
sanction	1125	5
slight	3344	8

Table 2: Data Attributes

One final and important note about the SENSEVAL taggings is that they are extremely fine. In particular, *band* had over 20 different possible senses defined, and the other words, although not as extreme, also had numerous possible senses. This clearly makes the tagging a more substantial challenge than in other connectionist approachs (Towell and Voorhees, 1998) that use a very limited number of possible sense tags. Table 2 reports the number of senses and vocabulary sizes for the words tested.

3.3 Learning Method

The learning method for a given network is the standard error backpropagation method used in teaching neural networks. After feeding the network a training input, the resulting output is compared to the expected output, error is computed, and weights and biases are adjusted accordingly.

One useful indicator for ending learning is the convergence of error of the network to a steady state. Using this as a basis, my network would train until error reached an asymptotic level. In general, this means the networks would learn for 15 to 20 epochs, seemingly quite fast. Given the size of the training set and the speed in training perceptrons, this is not altogether surprising.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College

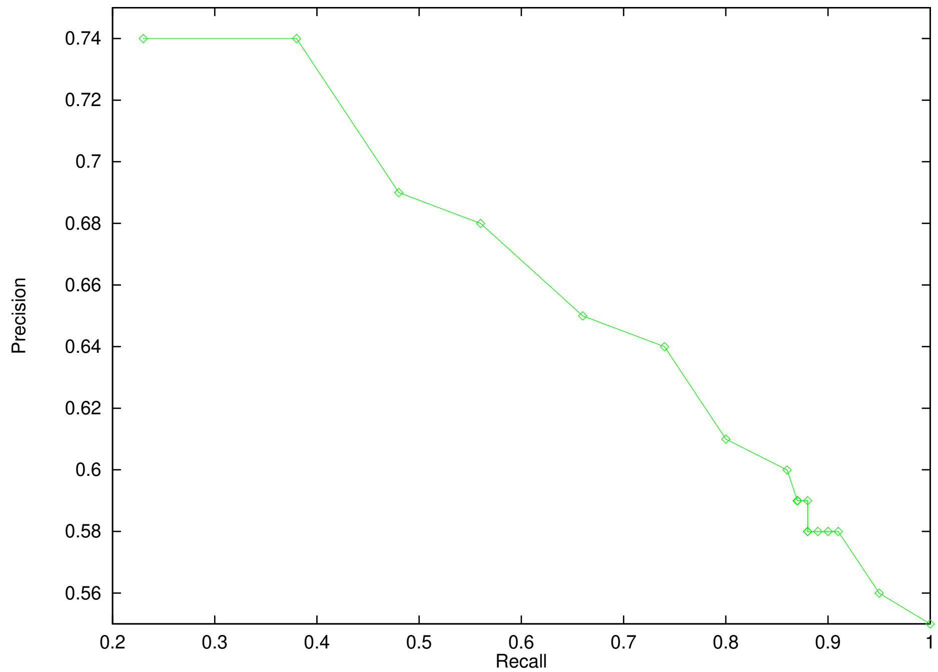

Figure 2: Precision-Recall Performance for *sanction*

word	*precision*	*F-Measure*
accident	70.4%	82.6%
band	64.4%	78.4%
brilliant	32.9%	49.5%
sanction	55.6%	71.5%
slight	69.7%	82.1%
average	58.6%	72.8%

Table 3: Performance with 100% Recall

4 Results

After training five different networks to disambiguate the target words, I was able to test the network's performance on the SENSEVAL test sets. The test sets were generally substantially smaller than the training sets. Initial testing results are reported in Table 3.

Again, it should be noted that the architecture of the neural net is set up to give 100% recall. For any given test sentence, a particular output node will be activated to a greater extent than any other output node. This 100% recall can be a problem, as most effective word sense disambiguators have a much lower recall. Nonetheless, performance is still quite good on all words except for brilliant. Also, the system's perfect recall leads to rather high F-Measures.

Given the potential problem that 100% recall is causing, a next step in testing was to try to lower the recall rate and raise precision. To do this, it was necessary to give the system the capability to tag a particular word as 'unknown.' I implemented this functionality by creating a threshold value to determine the certainty of the network's output. If the difference between the two highest activated output nodes was not greater than the threshold, then the system has an unacceptable degree of uncertainty about its output and chooses not to tag the word.

Adding this threshold technique for determining sense outputs, precision should be increased. To test the effectiveness of the threshold, I sampled a range of thresholds to plot the relation between recall and precision. We would expect to see an inverse relation. As the network stops tagging words that it is relatively unsure of, its recall falls but, since certainty of its taggings is higher, precision has likely increased. The test of this hypothesis is reported in Figure 2. Using the network trained to disambiguate the word *sanction*, increasing the certainty threshold causes a fall in recall and a rise in precision, the expected result.

Although this threshold permits the network to evaluate the certainty of various output taggings, the approach still has a few weaknesses. For one, the threshold must be assigned by an outside observer, and there appears to be no general rule for assigning the threshold. Instead, I sampled a variety of possible thresholds for the words and thereby selected thresholds that yielded seemingly reasonable results. It would be much more desirable to have the network generate its own thresholds and learn from them.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College

word	*precision*	*recall*	*F-Measure*
accident	74.0%	90.6%	81.5%
band	64.5%	99.0%	78.1%
sanction	63.8%	74.1%	68.6%
slight	78.3%	69.7%	73.7%
average	70.2%	83.4%	75.5%

Table 4: Performance with Lowered Recall

This well-documented relation between precision and recall suggests that better results can be achieved by lowering the sensitivity of the network's output. Using arbitrary thresholds, the networks' precision improved substantially, as shown in Table 4. It should be noted that brilliant has not been thoroughly tested with a threshold, due mostly to time constraints involved with finding thresholds for any particular net.

Unfortunately, the rise in precision in this approach was met with a more than proportional fall in recall. This fact can be seen by observing the change in F-Measures between the two tests. The average is slightly higher due to the absence of brilliant's results, but every individual F-Measure is worse than the tests with 100% recall. This drop in total system performance is certainly unexpected, and actually supports keeping the initial system intact and not using any threshold for determining certainty. One potential reason for this anomaly is the aforementioned arbitrary nature of the thresholds. A network that had incorporated certainty measures throughout learning would perhaps perform in a more expected fashion.

5 Discussion

Using the results from the SENSEVAL study, the connectionist approach described here stands up quite well. The state of the art for the statistical approaches used in the exercise is around 75-80% for both precision and recall(Kilgarriff and Rosenzweig, 2000). Although my system performs slightly worse on the five words I attempted, results are nonetheless quite comparable. A more apt comparison would clearly come from looking at the differing F-Measures for all the systems. Unfortunately, SENSEVAL results do not report this statistic for the evaluated systems. A rough calculation can be made, using the reported results of the best performers. If the best systems were between 75-80% in both precision and recall, then the system's F-Measures must be bounded between 75-80% as well. Using that as a comparison, my system performs admirably, with all tested words except brilliant having comparable results in the 100% recall test.

Although the network performed comparably on four of the five tested words, the results presented here are not a complete comparison to the SENSEVAL exercise. The full exercise had 35 words with 41 associated disambiguation tasks. These tasks included much more challenging tasks such as words with differing parts of speech and words with limited or no training data. The use of data with an ample training set might have unfairly influenced my system's performance. Nonetheless, my results are promising enough in general to prove that the connectionist approach can potentially compete with excellent statistical classifiers. Further work is certainly warranted to more generally test this approach's viability.

With regard to the specifics of the network performance, one important fact is the tendency for the networks to only focus on the most frequent senses. Even when presented with several senses, the network would usually ignore senses with very low frequency. In general, the network would only select the two or three most common senses as its chosen tags. One possible explanation for this behavior is the lack of hidden nodes. Hidden nodes would allow the network to develop a more nuanced approach to the context relevant for categorizing senses, and, as such, would be more likely to uncover the occurrences of less frequent words.

6 Future Work

The lack of hidden nodes provides an interesting arena for future research. The slow speed of network training prohibited an in-depth look at this current time, but I feel that future work could look into several interesting areas. As has been previously noted, fine taggings are likely to be better handled with hidden layers. Additionaly, hidden layers should be able to extract more intricate levels of meaning such as distinct phrases. Towell *et al* discuss this possibility, describing how diagnostic phrases such as 'stand in line' cannot be fully represented in a simple perceptron net based on topicality(Towell and Voorhees, 1998). A hidden layer would allow this sort of phrase to be characterized directly in one hidden node, albeit with that node probably handling several possible phrases from different contexts.

Another problem with the approach presented here is its reliance on having a unique network for every target word. A more robust possibility would be to create an enormous neural network that would incorporate the entire vocabulary from all the training sets as input nodes and additional input nodes specifying what word is currently ambiguous. The outputs for this network would be all the senses of all the words. A network architecture of this type is clearly enormous and is probably prohibitively costly to train or test, but nonetheless could potentially provide a much more general solution to the problem of word sense disambiguation.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College

7 Conclusion

This paper has presented a connectionist method of implementing word sense disambiguation. As this method is currently underexplored within the domain of natural language processing, this paper represents an important step in showing the feasibility of using neural networks for computational linguistic tasks. Further, the tests presented lend themselves to easy comparison to other systems' attempts at solving the same problem, as it utilizes the same testing and training corpora that were used in the SENSEVAL exercise.

My network has clearly demonstrated its ability to reasonably disambiguate a word given a sentence of context. Although the full range of SENSEVAL words was not fully tested, the results perform comparably with the systems that participated in the exercise, with the F-Measure of precision and recall averaging around 75%. Clearly, a fuller testing of all the words should provide a more complete analysis of the viability of a connectionist model.

Steps forward clearly include a deeper look into the potential advantages of using hidden nodes to allow increased generalization and more subtle analysis of context. Also, the automatic generation of certainty thresholds during training should permit the network to efficiently trade off between precision and recall. Nonetheless, this paper has successfully demonstrated that neural networks provide a reasonable framework for disambiguating word senses.

References

Kilgarriff Adam. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. Information Technology Research Institute Technical Report Series University of Brighton, U.K.

Kilgarriff Adam and Joseph Rosenzweig. English SENSEVAL: Report and Results. In Proceedings of the 2nd International Conference on Language Resources and Evaluation. 2000.

Mihalcea Rada and Dan I. Moldovan. 1999. An Automatic Method for Generating Sense Tagged Corpora. *AAAI IAAI*. 461-466.

Mooney Raymond J. 1996. Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Ed. Eric Brill and Kenneth Church. Association for Computational Linguistics. 82-91.

Ng Hwee Tou 1997. Getting Serious About Word Sense Disambiguation. *Tagging Text with Lexical Semantics: Why, What, and How? ANLP-97 Workshop*. Washington D.C.

Quinlan J. R. 1994. Comparing Connectionist and Symbolic Learning Methods. *Computational Learning Theory and Natural Learning Systems: Volume 1: Constraints and Prospects*. MIT Press. 445-456.

Towell Geoffrey and Ellen M. Voorhees 1998. Disambiguating Highly Ambiguous Words. *Computational Linguistics*. V. 24, N. 1 125-145.

Yarowksy David. 1993. One Sense Per Collocation. In *Proceedings, ARPA Human Language Technology Workshop*. Princeton, NJ. 266-71.

Yarowsky David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proceedings of the 33rd Annual Meeting of the Assocation for Computational Linguistics*. 189-96.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 78–83
Computer Science Department, Swarthmore College