Prepositional Phrase Attachment through Semantic Association using Connectionist Approach

Medimi Srinivas
Dept. of Computer Sc. and Engg.
Indian Institute of Technology, Bombay
Mumbai 400 076 India
msr@cse.iitb.ac.in

Pushpak Bhattacharyya
Dept. of Computer Sc. and Engg.
Indian Institute of Technology, Bombay
Mumbai 400 076 India
pb@cse.iitb.ac.in

Abstract

Determining the correct attachment site for Prepositional Phrase (PP) is one of the major sources of ambiguity in natural language parsing and analysis. In this paper, we describe a neural network based approach to prepositional phrase attachment for natural language text. Our approach disambiguates the attachment site for PP through semantic association among the constituents namely verb, noun and PP, using the WordNet semantic classes. It is essentially a corpus based approach. In most of previous corpus based statistical approaches, accurate estimation of probabilities was dependent on the data sufficiency in terms of size and coverage of the features. Moreover, rule-based systems are inappropriate for handling uncertain knowledge. Managing and maintaining rule based systems is also very difficult task and poses many problems. Our method, using the semantic class properties of words, reduces the lexical (word) level data sparseness problem. Neural networks are also very good in capturing the complex nature of semantic association among the words, and as a result capture the selectional restrictions. We have tested our method on Wall Street Journal corpus, and the experimental results show much better accuracy in PP attachment disambiguation and comparable to state-of-the-art approaches and the accuracy of the results shows the effectiveness of our approach.

1 Introduction

One of the central issues in natural language analysis is structural ambiguity resolution. Correct PP attachment determines the quality of structural analysis and as a consequence the semantic analysis of the natural language sentence. Contextual information and world knowledge are the two important requirements for successful resolution of prepositional phrase attachment ambiguity. Consider, for example, the following sentences:

1. *I purchased a toy for the baby*
2. *I purchased a toy for ten rupees*

In case of the first sentence, while computationally parsing, parser faces with two possible syntactic structures as follow:

1. I [purchased [$_{NP}$[$_{NP}$ a toy][$_{PP}$ for the baby]]]
2. I [purchased [$_{NP}$ a toy][$_{PP}$ for the baby]]

Human being can easily analyze the sentence as given in the parse 1. This is because he knows, using contextual information and world knowledge from his life time experience, that *toys are the playing material for children*. Automatic systems lack such common sense. Exploration of huge textual corpora can possibly partially substitute this world knowledge. Larger the size of corpus, more is the knowledge. However, in the corpus based approaches, there is always the problem of data sparsity for proper inferencing based on the lexical (word) statistics. Similarly, in the second sentence, the PP *for ten rupees* is attached to *purchase*.

Previous work on PP-attachment shows that building rule based systems using hand made rules are effective in small domain. However, it has its limitations in terms of handling uncertain knowledge. Further, managing and maintaining the rule based systems is a hard task. Any variation in the rule base may have unpredictable effect on the whole system or could affect other rules which may lead to rewriting the new set of rules. Due to these reasons, in recent years, corpus based methods such as automatic rule based methods and statistical methods and machine learning approaches are very successful. In the statistical and machine learning methods, the main limitation is data sparsity in terms of the lexical coverage and size. So, these systems can not handle the relationship not present in corpus. Solutions for countering data sparsity has been addressed in [Srinivas and Bhattacharyya, 2004a] and [Srinivas and Bhattacharyya, 2004b]. However, somehow, if all the lexical items can be categorized into a finite number of groups, then the problem of data sparsity can largely be eliminated in the sense that even if the lexical item is not present, based on its representative group correct attachment decision can be made. This reduces the problem to analyze and associate relationships among the limited number of group. This is what exactly has been described using connectionist approach in the subsequent sections of this paper. The reason for using neural network is its ability to capture complex association among the groups.

The road map of the paper is as follow: section 2 describes about the related work. Brief introduction about neural networks is given in section 3. In section 4, semantic groups of WordNet are briefly introduced. In section 5, a detail description about the neural network architecture for PP attachment is outlined. Then section6 explains about training of the neural networks and the results obtained. Section 7 analyzes the results obtained with other previous work. In

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 273–277.

section 8, possible improvements and future work are given. Section 9 concludes the paper followed by the references.

2 Related work

In the early works, two principles based on syntactic information are using either Minimal Attachment (MA) or Late Closure (LC) [Frazier, 1979]. This results in a wrong attachment in one of the following sentences:

1. *She eats pizza with a fork.*
2. *She eats pizza with sauce.*

Several semantic criteria have been worked out to resolve structural ambiguities. However, pinning down the semantic properties of all words is laborious and expensive, and is only feasible in a very restricted domain.

Subsequently, Altmann and Steedman (1988) [Altmann and Steedman, 1988] showed that the current discourse context is often required for disambiguating attachments. Recent works show that it is mostly sufficient to utilize lexical information (Brill and Resnik, 1994 [Brill and Resnik, In Proceedings of COLING 1994]; Collins and Brooks, 1995 [Collins and Brooks, 1995]; Hindle and Rooth, 1993 [Hindle and Rooth, 1993]; Ratnaparkhi et al., 1994 [Ratnaparkhi et al., 1994]). Most of the previous successful approaches to this problem have been statistical or machine learning and SVM based approaches. Pioneering research on corpus-based statistical PP-attachment ambiguity resolution has been done by Hindle and Rooth [Hindle and Rooth, 1993]. Hindle and Rooth were the first to show that a corpus based approach to PP-attachment ambiguity resolution can lead to good results.

Supervised methods by Ratnaparkhi et al. (1994) used a maximum entropy model considering only lexical information from within the verb phrase (ignoring N2). They experimented with both word features and word class features, their combination yielding 81.6% attachment accuracy. Another promising approach is the transformation-based rule derivation presented by Brill and Resnik in [Brill and Resnik, In Proceedings of COLING 1994], which is a simple learning algorithm that derives a set of transformation rules. Brill and Resnlk had reported 81.8% success of this method on 500 randomly-selected sentences.

Later, Collins and Brooks [Collins and Brooks, 1995] achieved 84.5% accuracy by employing a backed-off model to smooth for unseen events. They discovered that P is the most informative lexical item for attachment disambiguation and keeping low frequency events increases performance. The state of the art system is the supervised algorithm by Stetina and Nagao [Stetina and Nagao, 1997], that employs a semantically tagged corpus, and an unsupervised word-sense disambiguation algorithm with WordNet. Testing examples are classified using a decision tree induced from the training examples. They report 88.1% attachment accuracy approaching the human accuracy of 88.2% [Ratnaparkhi et al., 1994]. The unsupervised approach by Ratnaparkhi [Ratnaparkhi, 1998] achieves 81.9% attachment accuracy. The current unsupervised state of the art system, by Pantel et al [Pantel and Lin, 2000] which uses parser to collect the training data and its accuracy is 84.31%, which almost matches the supervised approach by Collins and Brooks [Collins and Brooks, 1995].

Most of the methods, however, suffer from a sparse data problem. The back-off model showed an overall accuracy of 84.5%, but the accuracy of full quadruple matches was 92.6%! Due to the sparse data problem, however, the full quadruple matches were quite rare, and contributed to the result in only 4.8% of cases. The accuracy for a match on three words was also still relatively high (90.1%), while for doubles and singles it dropped substantially [Collins and Brooks, 1995]. For instance, Bikel(2003) shows that the parser of Collins [Collin, 1999] is able to use bi-lexical word dependency probabilities to guide the parsing decision only 1.5% of the times, and for rest of times it backs off to condition one word on just phrasal and part-of-speech categories.

In general, there are two methods for establishing the relationship between the PP and the predicate, namely, methods using words (lexical preferences) and methods using semantic classes (selection preferences). The approaches to data sparsity reduction for PP attachment using WordNet have been described in [Srinivas and Bhattacharyya, 2004a] and [Srinivas and Bhattacharyya, 2004b], which show better results. These approaches use lexical preferences and replace words with WordNet synsets.

Lexical approaches for PP-attachment lead to data sparsity and involves handling of huge patterns. Where as, in semantic class/group approach, since it is assumed that all the words included in a group behave similarly, there are fewer classes to handle and moreover they help in statistical smoothing for even the missing words in the corpus. Some of the the class based methods [Resnik and Hearst, 1993; Ratnaparkhi et al., 1994; Brill and Resnik, In Proceedings of COLING 1994; Collins and Brooks, 1995] and [Li and Abe, 1995] have used WordNet [Miller et al., 1993] to extract word classes. However, the performance of the systems using only classes have not been good, but using words and classes are better than using only classes. This may be attributed to information loss due to semantic class consideration and less scope for discrimination. In fact our approaches in [Srinivas and Bhattacharyya, 2004a] and [Srinivas and Bhattacharyya, 2004b] performed better using the lexical and synsets. Another approach [Niemann, 1997], explains about PP attachment through semantic association and preferences using the WordNet noun hypernym groups and verb troponym trees. They show that prepositions, their objects and their attachment sites show a clear selectional preference patterns and however they tested in small data set. Since, many complex preference patterns are possible, we felt to capture such computation patterns using neural networks described here as they are very good at complex computation.

3 Neural Networks

Neural network is a network of the simple processing units called neurons. Each neuron can perform simple numerical calculations and produce the output using a nonlinear mapping function called activation function. The network of these simple processing units, along with their characteristic activation function, processing data in distributed fashion, can capture complex nature of mapping with good approximation. The functionality of neural network depend on the topology of the network, the connectivity, the activation function, the training algorithms and the learning parameters etc. Apart from the input and output layers, the hidden layers take vital role. One thing is very clear that given the neural network the performance of the network depend on the proper feature selection for the underlying problem. There are mainly three stages involved in the use of neural network namely, training, validation and testing. During training and validation stage, the neural network is trained with a large representative set of input for the problem concerned. During the training the network learns the mapping between the input and the output patterns and these mapping is stored in the form of link weights between the neurons in the inter and intra layers. Validation stage is to check the generalization of the network and if necessary is retrained with new set of data. In the test phase, the performance of neural network is tested with the unknown set of test data. The main advantage of neural networks is that they are able to approximate the known/unknown complex nonlinear mapping between the input and the output.

4 Semantic Groups and WordNet

WordNet [Miller et al., 1993] is a lexical database, which categorizes all words into several finite number of sets according to semantic concepts. Lexical data in the WordNet is organized by associating words with their antonyms, synonyms and hypernyms. In WordNet, nouns are grouped in 25 hypernym trees called a lexical group. The characteristic of each group(tree) is that for each member of a group there is another member that is its hypernym. At the top of the trees are '*unique beginners* that are hypernyms of all other members of the tree. For example,*dog* is a hypernym of *corgi*, they are both members of the *animal* group because the *animal* hypernym tree contains a branch that goes $corgi -> dog -> \ldots -> animal$. Verbs are grouped according to 15 *troponym* trees [Fellbaum, 1993]. For example, *limp* is a troponym of *walk* as to walk is to limp is to walk in a certain manner such that limping entails walking. WordNet has over 132,000 senses for nouns and verbs. Most objects and attachment sites can be categorized into a particular WordNet group which interpret as a semantic concept in our discussion.

5 Network Architecture for PP Attachment

Both the verbs and predicative nouns have their argument structures and impose selection restriction on PP. When both predicative nouns and verbs exist in a ambiguous frame structure *[V N1 P N2]*, PP is attached to to the predicative noun because predicative noun is nearer to PP. So, triplet frame structure of of the form *V P N2* and *N1 P N2* are very useful to capture the selection restriction of verb and noun on PP. Further, from neural network computational point of view, simpler structures are prefereable. With these two objectives in mind, we decompose the ambiguous structures *[V N1 P N2]* into one positive *[V P N2]* and one negative *[N1 P N2]* if the PP-attachment is V and decompose into opposite triplet frame like negative *[V P N2]* and positive *[N1 P N2]* if the PP-attachment is N1.

For preparing the training data, the nouns are represented in terms of the 25 hypernym groups for nouns in WordNet and similarly the verbs are represented in terms of 15 troponym trees. Each root node of a verb hierarchy belongs to only one lexicographical file. Verb synsets are divided into 15 lexicographical files on the basis of semantic criteria. We codified the 25 noun hypernym groups and 15 verb troponym groups. The hypernym groups are the feature for nouns and troponym groups are the features for verbs. So, a noun in represented as vector of length 25 and a verb is represented as a vector of length 15. Since each noun, usually have many senses, corresponding to the senses the respective hypernym positions in the vector are set to 1 and the remaining positions are set to 0. It is also possible that many senses of a word belong to the same hypernym group. In such cases the weight for that hypernym group position is incremented by the number of senses fall under that group category. for an example:

For the frame*[V N1 P N2]*: *give answer to problem*

Positive NPN:answer to problem: [1 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] [0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]

Negative VPN:give to problem: [2 1 2 11 0 2 1 6 0 2 0 15 2 0 0] [0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]

Similarly, for the frame *[V N1 P N2]*: *restricted subsidy for the students*

Positive VPN:restrict to student: [0 3 0 0 0 0 0 0 0 0 0 0 0 1 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0]

Negative NPN:subsidy to student: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0]

In our neural networks, there were three layers, one input layer consisting of 50 nodes for NPN, 25 nodes for each noun and 40 nodes for VPN, 15 nodes for verb and 25 nodes for noun. One hidden layer, which consists of nodes ranging from 10 to 25 nodes. There is only one unit in the output layer, which indicates '1' for attachment and 'O' for no attachment.

The Neural network architecture is feed forward network with one hidden layer and full connectivity between the input to hidden layers and hidden to output layers. The networks were trained with back propagation algorithms. The activation function used was logistic function.

6 Training and Experimental Results

For our experimental purpose, the ambiguous structures *[V N1 P N2]* were used, extracted from the Penn-Tree-Bank Wall Street Journal [Marcus and Santorini et al., 1993]. Each *[V N1 P N2]* was divided into *N1 P N2* positive and *V P N2* negative examples in case *P N2* is attached to *N1*, and *V P N2* positive and *N1 P N2* negative examples in case *P N2* is attached to *V*. For each preposition P (say 'to'), the positive and negative instance for each form of triplet frame structures *N1 to N2* and*V to N2* collected for training the neural networks for *N1 to N2* and *V to N2* respectively. Like wise different neural networks were trained for different *preposition*. The nouns and verbs were replaced by the noun group and verb group feature vector, using the WordNet. For some of the verbs and nouns WordNet does not have an entry. Proper name were substituted by the WordNet class *someone*, company name by business-organization, and the prefixed nouns such as *co-director* with *director*. Out of the extracted 20801 *[V N1 P N2]* structures for training, some tuples were discarded because of some of the words were not found in Word Net and could not be substituted. Finally, the number of *[V N1 P N2]* samples that obtained with features for training were 8487 with noun attachment and 7418 with verb attachment. Similarly, out of the 4039 *[V N1 P N2]* validation structures, finally obtained feature frames were 1774 with noun attachment and 1582 with the verb attachment. And out of the 3097 *[V N1 P N2]* test frames, finally obtained feature frames are 1489 with noun attachment and 1098 with the verb attachment.

As discussed in section3, application of neural network for PP attachment involves three phases, training, validation and testing. For training we used in total 15905 *[V N1 P N2]* frames, for validation in total 3356 *[V N1 P N2]* frames and for testing in total 2587 *[V N1 P N2]* frames. With a total of 21848 *[V N1 P N2]* frames, we used a cross-validation method as a measure of correct generalization. Instead of using the same set of data for training, validation and testing, we selected different set of data from the common pool of 21848, maintain the number of examples for training, validation and testing unaltered. We trained the networks for two runs, each run consisted of ranging from 100 to 500 epochs for different neural networks. The two runs were made with different starting random weights. Depending on the network, within each run after definite number of epochs the network was validated to ensure correct generalization. In each run the weights of the network having the smallest error with respect to the validation set were stored. The weights corresponding to the best results obtained on the validation test in the two runs were selected and used for testing purpose to evaluate the performance in the test set. After performing the each run we interchanged the validation and test data. Then the same process is repeated for training the network. The best network having the smallest error with respect to the validation set were stored. The whole experiment is repeated with a new set of training, testing and validation data sets. These data were collected from the same pool of 21848 *[V N1 P N2]* frames. The network was run with a varied random starting weight. We have not made any exhaustive exploration of the parameters for network training. Since we have trained two networks *V P N2* and *N1 P N2* corresponding to each preposition *P*, when test the attachment performance is evaluated based on the algorithm 1given below.

Table 1: Algorithm for PP attachment decision

Given a *[V N1 P N2]*,
IF *V P N2* network output indicates an attachment,
and *N1 P N2* network output also indicates also
an attachment, THEN
Attachment is to N1,
ELSIF *V P N2* network indicates an attachment THEN
Attachment site is V
ELSIF *N1 P N2* network indicates an attachment THEN
Attachment site is N1
ELSE Attachment site is N1

Table 2: Accuracy result and test size 'W' - Word only, 'C' - Class only,and 'W&C' - word and Class H&R=Handle and Roth, R&H=Resin and Hearst, RP=Ratnaparkhi et al. B&R=Brillo and Resin, C&B= Collins and Brooks, L&A= Li and Abe

Author	W	C	W & C	Classes	Test Size
H&R	80				880
R&H	81.6	79.3	83.9	Word Net	172
R&H			75	Word Net	500
RP	81.2	79.1	81.6	MIC	3097
B&R	80.8		81.8	Word Net	500
C&B	84.5				3097
L&A		85.8	84.9	Word Net	172

Some of the observations during experiments that, when we collect the positive and negative training examples, we should have a balanced proportion mix of the positive and negative ones. Otherwise, the network training is more biased to wards the majority examples. The performance of the trained neural networks on the test data is 85.9, compared with other systems. Our best over all result is 85.9. However, we believe that it can be further improved with further analysis on training and network structure.

7 Experimental Analysis

For our experiments, we used the standard Penn Tree bank Wall Street Journal data [Marcus and Santorini et al., 1993]. This is a standard benchmark data for PP attachment problem, which has been used by several previous researcher [Ratnaparkhi et al., 1994; Collins and Brooks, 1995; Stetina and Nagao, 1997]. For this test data, the attachment accuracy was 93.2% using the full sentence context, and 88.2% using only the 4-tuple [Ratnaparkhi et al., 1994] for the human evaluators.. Given the 4-tuple for attachment decision making, we can at best expect the performance of our system approaching 88.2%. The comparative performance of the different systems are given in table 2. The back-off model by Collins and Brooks[Collins and Brooks,

1995], is the best performing algorithm with an accuracy of 84.5% using the words alone on Penn Treebank Wall Street Journal data. Further better results have been reported on this test set by [Stetina and Nagao, 1997] with 88.1% and on other datasets by [Harabagiu and Pasca, 1999], but these algorithms use word sense disambiguation and named entity recognizers. Our best overall performance is 84.9, which is relatively better using the semantic association as given in table 2. More over, our test results are based on 11,886 examples. Comparing the test data size, overall performance of our system is comparable and competitive. Further, our results are obtained using only semantic class information. Using only class information with 100% coverage, by far the best attachment accuracy is 79.1%, though best performance using only class information is 85.8 by Li and Abe [Li and Abe, 1995], but for the test size of 172 only. So, considering the test data size and using only semantic approach, the performance of our systems is much better. This shows that neural networks are better in capturing the complex combination semantic association among the lexical items. We further see the scope of improvement on the results in the senses that we have not done exhaustive parameter exploration.

8 Future Work

Seeing the capability of neural network in capturing the complex semantic association, for the 3-tuples, we plan to do more exploration on the performance of the present networks and their training and improve the attachment accuracy further. Since, previous work on the literature claim that considering 4-tuple at a time give better results [Ratnaparkhi et al., 1994], and previous non-connectionist approaches applied can not capture the complex semantic association, we plan to investigate how can neural network be best utilized and we also want to investigate whether Neural network with 4-tuples will perform better, finding the regularities among the simultaneously presented more number semantic classes, or not.

9 Conclusion

In this paper, we have proposed a semantic approach for prepositional phrase attachment using neural networks. In our approach, words are represented in terms of semantic groups of WordNet depending on their senses. Neural networks are better in dealing with multidimensional inputs and very good in computing complex statistical functions implicitly capturing the semantic associations, and they are model free. The performance of the system can further be improved with exhaustive exploration and proper feature vector representation for lexical words.

References

G. Altmann and M. Steedman. 1988. Interaction with context during human sentence processing,. *Cognition*, page 30.

E. Brill and P. Resnik. In Proceedings of COLING, 1994. A rule based approach to pp attachment disambiguation. *In Proceedings of COLING,*.

M. Collin. 1999. Head driven statistical models for natural language parsing. *Doctoral dissertation, unuversity of pennsylvania,*.

M. Collins and J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. *In Proceedings of the Third Workshop on Very Large Corpora,*.

C. Fellbaum. 1993. English verbs as semantic net. *In Five Papers on WordNet*.

L. Frazier. 1979. On comprehending sentences: Syntactic parsing strategies. *Ph.D., University of Connecticut,*.

Harabagiu and Pasca. 1999. Integrating symbolic and statistical methods for prepositional phrase attachment. *Proceedings of FLAIRS*, pages 303–307.

H. Hindle and M. Rooth. 1993. Structural ambiguity and lexical relations. *Comp. Linguistics,*, 19(1):103–120.

H. Li and N. Abe. 1995. Generalising case frames using a thesarus and the mdl principle. *In the proceedings of the International Workshop on Parsing technology*.

M. Marcus and B. Santorini et al. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313–330.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1993. Introduction to wordnet:an online lexical database. *Anonymous FTP, Internet:clarity. prince.edu*.

Michael Niemann. 1997. Determining prepositional phrase attachment bysemantic assocaition and prefernces. *Hounours thesis, Deptt. of Linguistics and Applied Linguistics, Univ. of Melbourn*.

P. Pantel and D. Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextual similar words,. *ACL*, pages 101–108.

A. Ratnaparkhi, J. Reynar, and S. Roukos. 1994. Maximum entrophy model for prepositional phrase attachment. *In Procedings of the ARPA Workshop on Human Language Technology*.

A. Ratnaparkhi. 1998. Unsupervised statistical models for prepositional phrase attachment,. *In Proceedings of COLING-ACL98. Montreal, Canada, 1998*.

P. Resnik and M. Hearst. 1993. Syntactic ambiguity and conceptual relations. *In the proceedings of the ACL Workshop on Very Large Corpora*.

Medimi Srinivas and Pushpak Bhattacharyya. 2004a. Prepositional phrase attachment disambiguation using semantics: A supervised approach. *Fifth International conference on Knowledge Based Computer Systems (KBCS 2004), Hyderabad, INDIA*, 19–22 December.

Medimi Srinivas and Pushpak Bhattacharyya. 2004b. Unsupervised pp attachment disambiguation using semantics. *Third International conference on Natural Language Processing (ICON 2004), Hyderabad, INDIA*, 19–22 December.

J. Stetina and M. Nagao. 1997. Corpus based pp attachment ambiguity resolution with a semantic dictionary,. *In Proceedings of the Fifth Workshop on Very Large Corpora, pp. 66–80. Beijing and Hong Kong*.