A Hybrid Approach to Single and Multiple PP Attachment Using WordNet

Akshar Bharathi, Rohini U.[1], Vishnu P.[1], S.M. Bendre[2], and Rajeev Sangal[1]

[1] International Institute of Information Technology,
Hyderabad, India
rohini@research.iiit.net, vishnu@students.iiit.net, sangal@iiit.net
[2] University of Hyderabad, Hyderabad, India
bendre@iiit.net

Abstract. The problem of prepositional phrase attachment is crucial to various natural language processing tasks and has received wide attention in the literature. In this paper, we propose an algorithm to disambiguate between PP attachment sites. The algorithm uses a combination of supervised and unsupervised learning along with the WordNet information, which is implemented using a back-off model. Our use of the available sources of lexical knowledge base in combination with large un-annotated corpora generalizes the existing algorithms with improved performance. The algorithm achieved average accuracy of 86.68% over three test data sets with 100% recall. It is further extended to deal with the multiple PP attachment problem using the training based on single PP attachment sites and showed improvement over the earlier works on multiple pp attachment.

1 Introduction

Prepositional phrase (PP) attachment problem addresses structural ambiguity in natural language processing which is a major source of errors in parsing. The goal of PP attachment is to decide the attachment site of a given PP in the sentence. For example, consider the following sentences

a) *Mary ate the salad with a fork.*
b) *Mary ate the salad with croutons.*

In sentence a), the PP '*with a fork*' attaches to the verb *eat* rather than the noun *salad* and is called adverbial attachment. In sentence b), the PP '*with croutons*' attaches to *salad* rather than *eat* and is called adjectival attachment.

1.1 Related Work

The problem of disambiguation between the PP attachment sites has received wide attention in natural language processing. Many rule-based methods, statistical methods which comprise of supervised and unsupervised methods and hybrid methods are proposed for the ambiguity resolution.

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 211–222, 2005.

The prominent among the supervised methods which use annotated corpora for ambiguity resolution is the transformation-based approach by Brill and Resnik [1] with reported accuracy of 80% and the back-off approach to smoothen the probabilities of unseen attachments by Collins and Brooks [2] with reported accuracy of 84.5%. Ratnaparakhi et al [7] considered lexical information within the verb phrase and used maximum entropy model to achieve the accuracy of 81.6%. Stetina and Nagao [9] used the WordNet thesaurus and sense tagged corpus to achieve the accuracy of 88.1% using a decision tree for classification. Though supervised methods dominate unsupervised methods in performance and the accuracy achieved by Stetino and Nagao [9] is close to the human accuracy of 88.2% reported by Ratnaparakhi et al [7], the non-availability of large amount of annotated corpus is a serious limitation.

On the other hand, the unsupervised methods use un-annotated corpus and infer attachment site based on the lexical association. Hindle and Rooth [4] used the lexical associations of verbs and nouns by computing co-occurrence frequencies, which resulted in 82% correct attachments for a set of around 3000 test cases from the Penn Tree bank. Pantel and Lin [6] proposed an iterative approach using unsupervised training data. The algorithm uses contextually similar words derived from a collocation database and a corpus based thesaurus for classification with 84% accuracy. Zavrel et al [11] proposed a nearest-neighbor algorithm using memory based learning with an accuracy of 84.4%. Zhao and Lin [12] also used nearest-neighbor approach using various similarity measures and the algorithm achieved 86.5% accuracy using the cosine of mutual information as the similarity measure. Srinivas and Bhattacharya [8] extracted unambiguous data from raw corpus based on heuristics, expanded it using WordNet and used it as a training set, which yielded an accuracy of 83.86% on test data prepared by Ratnaparakhi et al [7]. Volk [10] combined the supervised and unsupervised approaches and used the back-off model for disambiguation on German corpus, achieving an accuracy of about 81% with a small annotated corpus of 10,000 sentences.

1.2 Proposed Approach

In this paper, we propose an approach which combines the strength of supervised and unsupervised approaches and also uses WordNet information whenever available to improve the disambiguation of attachment of a given PP. Our approach handles the problem of sparse data and the use of WordNet significantly differs from the earlier approaches (Stetino and Nagao [9]; Srinivas and Bhattacharya [8]).

The training phase consists of supervised and unsupervised learning from annotated and un-annotated corpora and computing supervised and unsupervised scores. The supervised scores for quadruplets, triplets and pairs are analogous to the scores considered by Collins and Brooks [2]. Further, information is iteratively extracted from the un-annotated corpus and is used to compute unsupervised scores for triplets and pairs. In addition, synonyms of verb and nouns present in a quadruplet are extracted from WordNet and their supervised and unsupervised scores are appropriately used to compute supervised and unsupervised WordNet scores respectively. All the calculated scores effectively give probability estimates of verb and noun attachments in the given situation. A convex combination of all these scores is used for disambiguation of the attachment of a given PP using a back-off model similar to Volk [10]. The approach achieves an accuracy of 86.5% on the test data in [7] containing 2998 quadruplets. The algorithm was also tested on two other data sets and achieved an average accuracy of 86.68% with 100% recall over all three data sets.

We further extend the algorithm to handle the problem of multiple PP attachment. A sentence often contains multiple prepositions, increasing the number of possible attachment sites and thus complicating the PP attachment problem further. For instance, out of 1223 sentences extracted from Penn Tree Bank, containing at least one preposition, all had two prepositions and 43% had three prepositions. The problem of multiple PP attachment has not received much attention in the literature. To our knowledge, there is a single reported attempt by Merlo et al [5] to disambiguate attachment sites in case of multiple PPs in a sentence. They used generalized back-off approach, re-using the single PP attachment training information for multiple PP attachment and achieved an accuracy of 84.3% for first PP, 69.6% for the second and 43.6% for the third PP on data extracted from Penn Tree Bank.

Our extended algorithm when run on the data extracted from Penn Tree Bank showed the accuracy of 86.5% for the first PP, 71.9% for the second and 58% for the third PP. The algorithm was also applied to the test data used by Merlo et al [5] and resulted in the accuracy of 88.99% for the first PP and 73.4% for the second PP. The noun belonging to the last PP in the sentence is not available in this test data and hence the accuracy of our algorithm for the third PP could not be calculated. The algorithm showed improvement over the accuracy achieved by Merlo et al for the first two PPs.

The rest of the paper is organized as follows. Section 2 briefly discusses the single and multiple PP attachment problem and Section 3 describes the training data. Section 4 details the supervised and unsupervised learning. Section 5 presents the disambiguation algorithm for single PP attachment and its evaluation. Section 6 discusses the extension of the single PP algorithm to multiple PP attachment problem and its evaluation and the conclusions are presented in Section 7.

2 Characterizing the PP-Attachment Problem

We first consider the single PP attachment problem. Given a sentence with a single PP, the sentence is typically reduced to a quadruplet (V, N, P_1, N_1) where V is the head verb, N is the head noun of the of the object of V, P_1 is a preposition and N_1 is the head noun of the PP (Ratnaparakhi et al [7]; Pantel and Lin [6]; Volk [10], among others). Thus, the PP attachment problem simplifies to the binary classification task of attaching the PP (P_1, N_1) to V (adverbial attachment) or to N (adjectival attachment).

In case a sentence contains multiple prepositions, the attachment sites for all the PPs need to be determined. An average English sentence usually contains multiple verbs as well as multiple PPs. For example, consider the following sentence from Penn Tree Bank,

ACET will shortly be opening a new office in the east end of London to serve clients in North and East London

which has three PPs and two verbs. The general structure of a sentence having multiple PPs can be represented by

$$V_a \;\; N_a \;\; P_1 \;\; N_1 \;\; P_2 \;\; N_2 \;\; \cdots \;\; V_g \;\; N_g \;\; P_k \;\; N_k \;\; \cdots$$

As a result, there is a multi-fold increase in the possible attachment sites for the second and subsequent PPs. In particular, for the sentence above, the representation is

$$V_a \;\; N_a \;\; P_1 \;\; N_1 \;\; P_2 \;\; N_2 \;\; V_b \;\; N_b \;\; P_3 \;\; N_3$$

and the possible attachments for the preposition phrases can be listed as

- $(P_1, N_1) \;\; \rightarrow \;\; V_a, \; N_a$
- $(P_2, N_2) \;\; \rightarrow \;\; V_a, \; N_a, \; N_1$
- $(P_3, N_3) \;\; \rightarrow \;\; V_a, \; V_b, \; N_2, \; N_b$

For instance, the possible attachment sites for the preposition '*of*' (P_2) in the sentence above are *open, office, end.* The increase in the number of possible attachment sites of subsequent PPs complicates the problem. Also, the presence of multiple verbs in the sentence further adds to the existing complexity. Note that the attachment ambiguity of the first PP (P_1, N_1) is the same as that of a single PP discussed earlier.

We assume that the attachment of a PP in a sentence is independent of the attachment of any other PP that occurs before or after it in the sentence. However, we use a few linguistic rules to rule out certain possible attachment sites, which are discussed in detail in Section 6.

3 Data Description

As mentioned earlier, our approach is a combination of supervised and unsupervised methods which uses two annotated and un-annotated corpora each.

The first annotated corpus[1] consists of 20,801 tagged quadruplets (V, N, P_1, N_1) from Wall Street Journal(WSJ), extracted from Penn Tree bank by the group at IBM. This corpus has been extensively used in earlier works on PP attachment (Ratnaparkhi et al [7]; Stetina and Nagao [9]; Zavrel et al [11], among others). The second annotated corpus consists of 1800 sentences extracted from texts G and H of British National Corpus (BNC)[2] and manually tagged by us. The first un-annotated corpus consists of 40,000 untagged sentences from WSJ . Our second un-annotated corpus consists of around 37 million words extracted from the texts A, B, C and D of BNC.

[1] ftp://ftp.cis.upenn.edu/pub/adwait/PPattachData

[2] http://www.natcorp.ox.ac.uk

For testing the single PP attachment algorithm, we considered three data sets. The first data set (data set I) is from the Penn Tree Bank (WSJ), consisting of 2998 tagged quadruplets collected by Ratnaparkhi [7]. The second data set (data set II) consists of manually tagged 1209 sentences which we extracted from Penn Tree Bank using TGrep2 [3]. For the third data set (data set III), we consider 4583 quadruplets corresponding to the first PP attachments from the test data used by Merlo et al [5][4]. The first test data for multiple PP attachment algorithm (data set IV) consists of 1223 manually annotated sentences extracted automatically from the Penn Tree Bank. In addition, we also tested the algorithm on the data (data set III) used by Merlo et al [5].

4 Learning from Training Data

In this section, we introduce the supervised and unsupervised scores based on supervised and unsupervised learning methods.

4.1 Supervised Learning

Initially, a few preprocessing steps such as morphing, converting all words to lower case, replacing numbers and years by a common token 'NUMBER' etc were carried out on the annotated corpora. The frequencies of quadruplets, triplets, pairs and prepositions for noun and verb attachment were calculated. Based on these frequencies, we compute the following supervised and unsupervised scores analogous to [2].

$$DV = f(0, V, P_1, N_1) + f(0, N, P_1, N_1) + f(0, V, N, P_1)$$
$$DN = f(1, V, P_1, N_1) + f(1, N, P_1, N_1) + f(1, V, N, P_1)$$

where f stands for frequency of occurrence of the triplet in data, 0 stands for verb attachment and 1 for noun attachment. The supervised verb and noun scores for triplets are

$$V_{sup_N}(V, P_1, N_1) = \frac{DV}{DV + DN}, \quad N_{sup_V}(N, P_1, N_1) = \frac{DN}{DV + DN} \qquad (1)$$

The subscripts N and V in sup_N and sup_V in (1) above stand for the exact N and V present in the quadruplet, indicating the dependence of the scores on N or V respectively.

Similarly, we compute the scores for the pairs (V, P_1), (N, P_1) and preposition P_1 which are denoted by $V_{sup_{N,N_1}}(V, P_1), N_{sup_{V,N_1}}(N, P_1)$ and $V_{sup}(P_1)$, $N_{sup}(P_1)$ respectively.

[3] http://tedlab.mit.edu/ dr/TGrep2/
[4] http://www.latl.unige.ch/personal/cathy_f.html

4.2 Unsupervised Learning

In order to learn from un-annotated corpus, we carried out an iterative approach analogous to Pantel and Lin [6] with modified scores. Each sentence of the un-annotated corpus was parsed using Minipar[5] ignoring the PP attachments. The parsed sentences were used to extract the quadruplets of the form (V, N, P_1, N_1) for every existing PP (P_1, N_1). Each of the extracted quadruplet was reduced to two triplets (V, P_1, N_1) and (N, P_1, N_1) and an initial value of 0.5 was assigned to each triplet. The value of 0.5 can be interpreted as the initial probability that the PP (P_1, N_1) gets a verb or a noun attachment. If only one triplet is extracted from a parsed sentence, a value of 1 is assigned to it. Let $V_{value}(V, P_1, N_1)$ be the sum of the initial values assigned to (V, P_1, N_1) over the entire corpus and similarly we compute $N_{value}(N, P_1, N_1)$. For a specific triplet (V, P_1, N_1), we define proportion as

$$Prop(V, P_1, N_1) = \frac{V_{value}(V, P_1, N_1)}{\sum_{v_i} V_{value}(v_i, P_1, N_1)} \tag{2}$$

where v_i ranges over all verbs occurring with the PP (P_1, N_1) in the un-annotated corpus. $Prop(V, P_i, N_1)$ is an empirical estimate of the probability that the PP (P_1, N_1) occurs with this specific verb V. These proportions are analogous to the frequencies defined by Pantel and Lin [6], but unlike them, we retain P_1 and N_1 in the computations. We believe that N_1 provides context information and as pointed out by Collins and Brooks [2], the preposition P_1 plays a major role in deciding the attachment.

Starting with the initial value in (2), we iteratively modify $Prop$ for V by modifying V_{value} to $Prop(V, P_1, N_1) + \sum_{n_i} Prop(V, P_1, n_i) + \sum_{v_i} Prop(v_i, P_1, N_1) + \sum_{v_i, n_i} Prop(v_i, P_1, n_i)$. Effectively, this is a back-off smoothing to get better expectations of V_{value} from the unsupervised corpus. Note that the computation of V_{value} is the Expectation-step and estimating probabilities through $Prop$ is the Maximization-step of the EM algorithm. By using back-off smoothing of V_{value} in between, we modify the expectations computed in the E-step. The iterations are continued till the value of $Prop(V, P_1, N_1)$ stabilizes. The stabilized value gives a smoothed estimate of the probability mentioned earlier. $Prop(N, P_1, N_1)$ is computed on the same lines using N_{value}.

From the $Prop$ values of triplets thus obtained, we calculate the unsupervised scores for triplets as,

$$V_{unsup}(V, P_1, N_1) = \frac{Prop(V, P_1, N_1)}{\sum_{v_i} Prop(v_i, P_1, N_1)} \tag{3}$$

$$N_{unsup}(N, P_1, N_1) = \frac{Prop(N, P_1, N_1)}{\sum_{n_i} Prop(n_i, P_1, N_1)} \tag{4}$$

where the sum in the denominator of (3) is over all the verbs which co-occur with (P_1, N_1) in the training set. The unsupervised scores for pairs are calculated on the same lines and we skip the details here..

---

[5] http://www.cs.ualberta.ca/ lindek/minipar.htm

The resulting database at the end of the learning stage consists of triples and pairs with their corresponding supervised and unsupervised scores.

5 Single PP Attachment

We now propose an algorithm for the disambiguation of attachment sites for a single PP. As mentioned earlier, the algorithm combines the information learnt from supervised and unsupervised learning with that of WordNet.

Methods incorporating WordNet in PP attachment algorithm have been proposed earlier by Stetina and Nagao [9] for sense disambiguation in constructing a decision tree for PP attachment disambiguation, and Srinivas and Bhattacharya [8] to expand the training set size by replacing each word in the training set by its synonyms. As mentioned earlier, our use of WordNet significantly differs from the above two approaches.

For disambiguation, if the given quadruplet (V, N, P_1, N_1) is present in the annotated corpus, it is assigned the attachment given by the annotated corpus. Otherwise the sets of synonyms are extracted from WordNet for each of V, N and N_1, which we denote by $\mathcal{C}_V$, $\mathcal{C}_N$ and $\mathcal{C}_{N_1}$ respectively. Using the quantities defined in (1)-(4) for the two triplets (V, P_1, N_1) and (N, P_1, N_1), we define WordNet scores for V and N as follows

$$WV_i(V, P_1, N_1) = \begin{cases} \sum_{v_i \in \mathcal{C}_V} \sum_{n_i \in \mathcal{C}_N} \sum_{n1_i \in \mathcal{C}_{N_1}} \frac{g(V_{sup_{n_i}}(v_i, P_1, n1_i))}{|\mathcal{C}_V| * |\mathcal{C}_N| * |\mathcal{C}_{N_1}|} & \text{if } i = sup \\ \sum_{v_i \in \mathcal{C}_V} \sum_{n1_i \in \mathcal{C}_{N_1}} \frac{g(V_{unsup}(v_i, P_1, n1_i))}{|\mathcal{C}_V| * |\mathcal{C}_{N_1}|} & \text{if } i = unsup \end{cases} \tag{5}$$

$$WN_i(N, P_1, N_1) = \begin{cases} \sum_{v_i \in \mathcal{C}_V} \sum_{n_i \in \mathcal{C}_N} \sum_{n1_i \in \mathcal{C}_{N_1}} \frac{g(N_{sup_{v_i}}(n_i, P_1, n1_i))}{|\mathcal{C}_V| * |\mathcal{C}_N| * |\mathcal{C}_{N_1}|} & \text{if } i = sup \\ \sum_{v_i \in \mathcal{C}_V} \sum_{n1_i \in \mathcal{C}_{N_1}} \frac{g(N_{unsup}(n_i, P_1, n1_i))}{|\mathcal{C}_{\mathcal{V}}| * |\mathcal{C}_{N_1}|} & \text{if } i = unsup \end{cases} \tag{6}$$

If any of the triplets is not present in the training corpus, the score is taken to be zero. The function g used in the scores in (5) and (6) is an appropriate weight function. In particular, one can consider binary functions of the type $g(V_{sup_{n_i}}) = 1$ if $V_{sup_{n_i}} > N_{sup_{v_i}}$ and 0 otherwise. We consider the convex combinations of the scores introduced in (1) to (6) above to define the final scores which are used for the disambiguation and are given by

$$FinalVScore_i(V, P_1, N_1) = \alpha\ WV_i(V, P_1, N_1) + (1 - \alpha)\ V_i(V, P_1, N_1) \tag{7}$$

$$FinalNScore_i(N, P_1, N_1) = \alpha\ WN_i(N, P_1, N_1) + (1 - \alpha)\ N_i(N, P_1, N_1) \tag{8}$$

where α is an appropriately chosen value between 0 and 1 and i is sup or $unsup$ as the case may be. For the pairs (V, P_1), (N, P_1) and (P_1, N_1) extracted from the given quadruple (V, N, P_1, N_1), the WordNet scores and the final scores are calculated on similar lines. The details of the score calculations for the pairs are presented in Appendix.

To summarize, the disambiguation algorithm is as follows. Given a quadruplet, if the preposition is *'of'*, a noun attachment is assigned irrespective of the verb. Next, if the quadruplet exits in the supervised data the tagged attachment is assigned. Else, a back-off model is employed using the final supervised scores first and then unsupervised scores if needed, at each stage of the model. Also, at each stage, if $FinalVScore$ is larger than $FinalNScore$, verb attachment is assigned and noun attachment otherwise. If no attachment is assigned up to the pair stage, the algorithm goes to Level B, where the site is assigned based on the attachment given to the preposition in the annotated corpus. If this leads to a tie, the algorithm goes to Level C, where the default attachment of noun is given to the PP, since it has been reported that choosing noun as the attachment site yields an accuracy of 58.96% [6].

The combination of information from corpora and WordNet used in the algorithm also takes care of the sparse data. We believe that this kind of combination of information helps in disambiguating the attachment even when there is a narrow difference in the noun and verb attachment scores. In Table 1 below, we present the number and percentage of quadruplets identified and the accuracy of the algorithm at each stage of the algorithm for the three test data sets I, II and III described in Section 3. To make certain that our test data sets II and III are not overlapping with the training data set, we did not consider the supervised quadruplets identified by the algorithm for calculating the precision for these two data sets. Hence, the precision reported for data set II is for 863 sentences and that for data set III is for 2481 sentences. The precision increases by about 2-3% when supervised quadruplets are considered.

Table 1. Single PP Attachment : Stage-wise Results

Data set Size	I (Ratnaparkhi) 2998		II (WSJ) 1209		III (Merlo et al) 4583	
Stage	Identified (%)	Accuracy (%)	Identified (%)	Accuracy (%)	Identified (%)	Accuracy (%)
'*of*'(noun)	29.45	95.1	42.34	99.21	31.96	99.86
Sup Quad	2.33	85.7	28.61	87.57	45.86	89.72
Sup Trip	18.21	84.2	14.39	79.31	8.42	83.28
UnSup Trip	2.33	81.41	2.56	61.29	3.09	64.08
Sup Pair	36.45	82.43	5.54	67.16	5.73	82.5
Unsup Pair	3.73	72.32	2.73	60.60	1.57	72.22
Default(B, C)	7.47	65.1	3.80	34.78	2.68	54.47

Table 2. Single PP Attachment : Overall Results

	Data set I		Data set II		Data set III	
Accuracy	Precision	Recall	Precision	Recall	Precision	Recall
Without Default	**86.32**%	92%	**89.35**%	94.66%	**90.79**%	95.04%
With Default	84.6%	**100**%	86.44%	**100**%	88.99%	**100**%

Table 2 below gives overall precision for the three data sets with and without default stage of level B and C. As anticipated, the precision increases with lower recall.

The average precision over all three data sets is 86.68 % for 100% recall and removing the default stages B and C from the algorithm increases the average precision to 88.82% with 93.9% recall. Though the precision for the data set III is surprisingly high, the precision for the data set I did not surpass the human accuracy of 88.2% reported by Ratnaparkhi et al [7].

6 Extension to Multiple PP Attachments

In this section, we discuss the extension of the proposed single PP attachment algorithm to handle the multiple PP attachment ambiguity. We assume that the decision of the attachment site of one PP in a sentence is independent of the attachment sites of the other PPs in the same sentence. This assumption allows us to use the single PP training data for multiple PP attachment problem. It also enhances the performance of the algorithm by reducing the possible attachment sites for the PPs. Before discussing the algorithm, we present the rules used for reducing possible attachment sites of PPs.

We first resolve the ambiguity among multiple verbs by using clause boundary information, since a preposition can attach only to elements within a clause. The clause boundary information is extracted from the phrase structure tree given by Collins parser. The accuracy of clause boundary identification of Collins parser is reported to be 85% ([3]). Given a test sentence, we identify the clause in which a preposition falls and rule out the other verbs as possible attachment sites, reducing the possibility of multiple verbs as attachment sites. For instance, the clause boundaries for the example sentence of Section 2 are

ACET [will shortly be opening a new office in the east end of London [to serve clients in North and East London]]

The second PP '*in North and East London*' falls in the clause headed by the verb '*serve*', which rules out the verb '*open*' as a possible attachment site.

The preposition may still have a verb and multiple nouns as its attachment sites within the same clause. Though we assume independence of PP attachments, we make use of linguistic knowledge to rule out certain attachment sites. We apply a rule which does not allow edges corresponding to the attachments to cross. For instance, in a structure of the type

$$V_a \quad N_a \quad P_1 \quad N_1 \quad P_2 \quad N_2$$

if (P_1, N_1) attaches to V_a then (P_2, N_2) cannot attach to N_a. In the above example, if the PP (P_1, N_1) '*in the east end*' attaches to '*open*', then the PP '*of London*' can not attach to '*office*'. Though the above rule further reduces the possible attachment sites, the ambiguity in the attachment sites of the preposition still persists.

As a first step towards using the single PP algorithm, we construct all possible quadruplets using the verb and available nouns with this PP. For instance, if the possible attachment sites of (P_2, N_2) are V_a, N_a and N_1, then the quadruplets constructed are (V_a, N_a, P_2, N_2), (V_a, N_1, P_2, N_2).

Given any multiple PP sentence, the attachment site of the first preposition is decided based on the algorithm described in Section 5. For each of the subsequent PPs, we first run the single PP attachment algorithm for all the constructed quadruplets. If all the quadruplets give verb as its attachment, adverbial attachment is assigned for the PP. If the attachments given by quadruplets are contradictory, we compute $\lambda Score$ for each of the quadruplet, defined as

$$\lambda Score(V, N, P_1, N_1) = \beta\ E_1 + (1 - \beta)\ E_2$$

where β is an appropriately selected normalizing constant between 0 and 1, E_1 is $FinalVScore_i$ for the quadruplet (V, N, P_1, N_1) and E_2 is $FinalNScore_i$ for the quadruplet (V, N, P_1, N_1) defined in (7) and (8), where i is sup or $unsup$, as the case may be. We pick the quadruplet with highest $\lambda Score$ and the PP attachment given by this quadruplet is assigned to the PP.

The above approach was tested using data set IV (Section 3) of 1223 sentences from WSJ, extracted from Penn Tree Bank. All sentences have at least two PPs and 43% of them have three PPs. As mentioned in Section 3, we considered data set III [5] of tuples extracted from 4583 sentences consisting of two or three PPs. The noun belonging to the last PP in the sentence is not available in this test data. Hence only those tuples with two or more PPs could be used to test the accuracy of the attachment of first PP. Similarly, to test for accuracy of the attachment of the second PP, we had to use tuples from the sentences with three PPs. Since all the tuples had a maximum of three PPs only, the accuracy of our algorithm for the attachment of the third PP could not be calculated for data set III. Table 3 presents the performance of our algorithm on data sets IV and III.. Note that analogous to the single PP case, the recall including the default

Table 3. Multiple PP Attachment Results

Data set	IV (WSJ)			III (Merlo et al)		
	PP_1	PP_2	PP_3	PP_1	PP_2	PP_3
Total no. of PPs	1223	1223	523	2581	430	–
Correct	1058	880	303	2208	316	-
Precision	**86.5%**	**71.9%**	**58%**	**88.99%**	**73.4%**	–

is 100% here. The accuracy of PP_1 is similar to the accuracy reported for the single PP attachment (Tables 1 and 2). For the first and the second PP of the test data III, the algorithm achieved 88.99% and 73.4% accuracy respectively. The corresponding accuracies using the algorithm by Merlo et al [5] are 84.3% and 69.6% respectively. The reported accuracy for the third PP is 43.6% and we believe that with the availability of the noun in the third PP, our algorithm would have achieved higher accuracy.

7 Conclusions

In this work, we have looked at the problems of both single PP attachment as well as multiple PP attachment. We combine both supervised and unsupervised methods along with the WordNet information in our algorithm. Our use of WordNet information significantly differs from the earlier approaches using WordNet. The training data from annotated corpora contains quadruplets with PP attachment information, whereas in case of unsupervised training, the data consists of co-occurrence information about the PP and its attachment sites. The training data sets have been extracted from WSJ as well as BNC. Two existing test data sets ([7], [5]) and a third data set extracted from WSJ were used for evaluating the accuracy of the algorithm. The precision of 86.32% with a recall of 92% was achieved on Ratnaparkhi's dataset [7] consisting of 2998 quadruplets. We achieved the average precision of 86.68% with 100% recall and the average of precision of 88.82% with 93.9% recall on the three data sets.

Multiple PP attachment problem was reduced to a problem of stepwise attachment of PPs from left to right within a clause. As a result, the single PP attachment algorithm can be extended to disambiguate each PP attachment site. Based on the attachment decisions for the earlier PPs, certain later PP attachments are ruled out because of non-crossing of attachments. On the test data consisting of 1223 sentences extracted from WSJ the algorithm achieved the precision of 86.5% for the first PP and 71.9% and 58% respectively for subsequent PPs with 100% recall. The algorithm was also tested on the data set used in [5] and showed improvement over the accuracy by the earlier algorithm. Annotated corpus plays an important role in reaching these levels of accuracy and a larger annotated corpus would help in improving this accuracy.

In conclusion, the algorithm shows significant improvement over earlier approaches to single and multiple PP attachment problem. Using thesaurus in place of WordNet is likely to improve the performance further since a thesaurus typically gives larger number of synonyms and does not provide subdivision of senses as fine as WordNet. This possibility is currently being investigated.

While considering second and subsequent PPs in a multiple PP sentence, the quadruplets that are formed do not have information that the noun at a possible attachment site may or may not be an object of the verb. For example, consider the sentence *He put the book on flowers on table.* The PP '*on flowers*' attaches to the noun '*book*' where as '*on the table*' attaches to the verb '*put*'. With the verb like '*put*', both prepositions '*on*' are highly likely to be attached to '*put*' because of the mandatory requirement of the verb frame. Thus the availability of verb frame information will make the task of PP attachment easier. Incorporation of such syntactic information would require a change in the algorithm and may be attempted in the future. In the presence of coordinate structure of two nouns or nous or verbs, our algorithm uses only the last element (noun or verb) in the coordinated structure. Handling of coordinate structure will also be pursued in future.

References

1. Brill, E., Resnik, P.: A Rule based Approach to PP Attachment Disambiguation. In Proceedings of COLING (1994)
2. Collins, M., Brooks, J.: PP Attachment Through a Backed-Off Model. In Proceedings of the Third Workshop on Very Large Corpora (1995)
3. Collins, M.: Head-Driven Statistical Models for Natural Language Processing. Ph.D. Thesis, University of Pennsylvania (1999)
4. Hindle, H., Roth, M.: Structural Ambiguity and Lexical Relations. Computational Linguistics **19** (1993) 103–120
5. Merlo, P., Crocker, M., Berthouzoz, C.: Attaching Multiple Prepositional Phrases:Generalized Backed-off Estimation. In Proceedings of the $2^n d$ conference on Empirical Methods in Natural Language Processing(EMNLP-2) (1997) 149–155
6. Pantel, P., Lin, D.: An Unsupervised Approach to Prepositional Phrase attachment using contextually similar words. In Proceedings of ACET(2000) 101–108
7. Ratnaparkhi, A., Reynar, J., Roukos, S.: Maximum Entropy Model for Prepositional Phrase Attachment. In Proceedings of the ARPA Workshop on Human Language Technology (1994)
8. Srinivas, M., Bhattacharya, P.: Unsupervised PP Attachment Disambiguation Using Semantics. In *Recent Advances in Natural Language Processing*, Allied Publications, New Delhi, India (2004)
9. Stetina, J., Nagao, M.: Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In Proceedings of the Fifth Workshop on Very Large Corpora (1997) 66–80
10. Volk, M.,: Combining unsupervised and supervised methods for PP attachment disambiguation. In Proceedings of COLING-2002, Taipei (2002)
11. Zavrel, J., Daelemans, W.: Memory-based Learning: Using similarity for smoothing. In Proceedings of 35th annual meeting of the ACL, Madrid (1997)
12. Zhao, S., Lin, D.: A Nearest Neighbour Method for Resolving PP-Attachment Ambiguity. In Proceedings of the First International Joint Conference on Natural Language Processing (2004)

Appendix . WordNet Scores for Pairs

The initial scores for verb and noun based on annotated corpus are $V_{sup_{N,N_1}}(V,P_1)$, $N_{sup_{V,N_1}}(N,P_1)$ and those using un-annotated corpus are $V_{unsup}(V,P_1)$ and $N_{unsup}(N,P_1)$. WordNet scores for pairs are calculated analogous to those in (5) and (6), given by

$$WV_i(V,P_1) = \begin{cases} \sum_{v_i \in \mathcal{C}_V} \sum_{n_i \in \mathcal{C}_N} \sum_{n1_i \in \mathcal{C}_{N_1})} \frac{g(V_{sup_{n_i,n1_i}}(v_i,P_1))}{|\mathcal{C}_V|*|\mathcal{C}_{N_1}|*|\mathcal{C}_N|} & \text{if i = sup} \\ \sum_{v_i \in \mathcal{C}_V} \frac{g(V_{unsup}(v_i,P_1))}{|\mathcal{C}_V|} & \text{if i = unsup} \end{cases}$$

$$WN_i(N,P_1) = \begin{cases} \sum_{v_i \in \mathcal{C}_V} \sum_{n_i \in \mathcal{C}_N} \sum_{n1_i \in \mathcal{C}_{N_1}} \frac{g(N_{sup_{v_i,n1_i}}(n_i,P_1))}{|\mathcal{C}_N|*|\mathcal{C}_{N_1}|*|\mathcal{C}_N|} & \text{if i = sup} \\ \sum_{n_i \in \mathcal{C}_N} \frac{g(N_{unsup}(n_i,P_1))}{|\mathcal{C}_N|} & \text{if i = unsup.} \end{cases}$$

Similar to final triplet scores in (7) and (8), the final pair scores are

$$FinalVScore_i(V,P_1) = \alpha\ WV_i(V,P_1) + (1-\alpha)\ V_i(V,P_1)$$
$$FinalNScore_i(N,P_1) = \alpha\ WN_i(N,P_1) + (1-\alpha)\ N_i(N,P_1)$$