A Flexible Unsupervised PP-Attachment Method Using Semantic Information

Srinivas Medimi
Deptt. of Computer Sc. & Engg.
Indian Institute of Technology Bombay
Mumbai, INDIA, 400076
msr@cse.iitb.ac.in

Pushpak Bhattacharyya
Deptt. of Computer Sc. & Engg.
Indian Institute of Technology Bombay
Mumbai, INDIA, 400076
pb@cse.iitb.ac.in

Abstract

In this paper we revisit the classical NLP problem of prepositional phrase attachment (PP-attachment). Given the pattern $V-NP_1-P-NP_2$ in the text, where V is verb, NP_1 is a noun phrase, P is the preposition and NP_2 is the other noun phrase, the question asked is *where does* $P-NP_2$ *attach:* V *or* NP_1? This question is typically answered using both the word and the world knowledge. Word Sense Disambiguation (*WSD*) and Data Sparsity Reduction (*DSR*) are the two requirements for PP-attachment resolution. Our approach described in this paper makes use of training data extracted from raw text, which makes it an *unsupervised* approach. The *unambiguous* $V-P-N$ and N_1-P-N_2 tuples of the training corpus TEACH the system how to resolve the attachments in the ambiguous $V-N_1-P-N_2$ tuples of the test corpus. A graph based approach to word sense disambiguation (*WSD*) is used to obtain the accurate word knowledge. Further, the data sparsity problem is addressed by (i) detecting synonymy using the wordnet and (ii) doing a form of inferencing based on the matching of Vs and Ns in the unambiguous patterns of $V-P-NP$, NP_1-P-NP_2. For experimentation, Brown Corpus provides the training data and Wall Street Journal Corpus the test data. The accuracy obtained for PP-attachment resolution is close to 85%. The novelty of the system lies in the flexible use of WSD and DSR phases.

1 Introduction

Correct PP-attachment is essential for syntactic and consequent semantic analysis of a sentence. For example: in the sentence *I lifted the girl with a crane*, *with crane* is the prepositional phrase (PP), and the PP-attachment could either be *lifted with a crane* (*machine sense of crane*) or *girl with a crane* (*bird sense of crane*). Word sense disambiguation and PP-attachment mutually affect each other for correct and unambiguous assignment of senses and PP-attachment respectively, as is evident from this example. Many previous researches [Ratnaparkhi, 1998; Donald Hindle and Rooth, 1993] have not considered the properties of nouns inside the PPs, but these make a difference. For example: In sentence, *I ate rice with salad/spoon*, the attachments are *rice with salad* and *ate with spoon*. In our approach, we consider the contribution of noun within PP for PP-attachment. Further, since most of the previous methods have focused only on $V-N_1-P-N_2$ structures, they can not consider the attachments of far away PPs, which are mostly adjuncts. We deal with such situation in our unsupervised approach.

Even with large corpora, the biggest challenge is data sparsity. We propose a simple, yet effective method of *Data Sparsity Reduction (DSR)*- using WordNet[1]. *DSR* helps smooth (generalize) the low probability counts. The correct PP-attachment is basically determined by the semantic property of the lexical items in the context of the preposition. We use an iterative graph-based unsupervised Word Sense Disambiguation (WSD) method (simlar to [Mihalcea, 2005]), which exploits the global semantic dependency and interaction of the word senses and ranks the senses of each word. This helps in correct PP-attachment.

The remainder of the paper is organized as follow: Section 2 describes a graph based iterative unsupervised Word Sense Disambiguation method. Section 3 explains the data sparsity reduction process. Section 4 details the unsupervised PP-Attachment method. Section 5 presents the experimental results and Section 6 concludes the paper.

2 Word Sense Disambiguation for PP-attachment

In this section, we describe a Word Sense Disambiguation (WSD) method in the context of Prepositional Phrase Attachment (PP-Attachment). The approach is an iterative graph based algorithm which performs random walk on the sense dependency graph of the words involved in the context. Our approach to word sense disambiguation for PP-attachment is based on two hypotheses: (1) *Prepositions are semantic carriers. Most of the semantic relations are derived in association with prepositions* (2) *Exploitation of the global semantic dependency among the words (preferably in the proximity of a preposition) will help WSD*. During the completion of this module of our work, a similar work on a generic graph-based approach for sequence data labeling and its application for WSD was published [Mihalcea, 2005].

[1] wordnet.princeton.edu

2.1 Graph Based Algorithm for WSD

We perform a random walk on the graph for each context, *i.e*, a sequence of words that are supposed to disambiguate each other. For us the sequence is $V - N_1 - P - N_2$. Since P is not a content word, the interaction is basically among V, N_1 and N_2. A typical graph for a $V - N_1 - P - N_2$ sequence is shown in figure 1. We formally define the graph as follow:
Let the given sequence of words be $W = w_1, w_2, w_3, ..., w_n$, and let each word w_i has N_{w_i} senses. Assume the senses for the word w_i are $S_{w_i} = \{s^1_{w_i}, s^2_{w_i}, s^3_{w_i}, ..., s^{N_{w_i}}_{w_i}\}$. We construct a labeled graph $G = (V, E)$ in which each vertex represents a word sense and is labeled with the sense number S_{w_i}. In G, there is a vertex v for every possible sense of a word $\{s^j_{w_i},\ i = 1..n,\ j = 1..N_{w_i}\}$. The dependency edges between a pair of vertices are labeled with the sense dependency representing the sense similarity of the concepts, which will be explained later. The word senses of the same word are not connected within themselves. Rather they are connected to labeled sense vertices of other words. However, not all labeled pairs can be related by dependency.

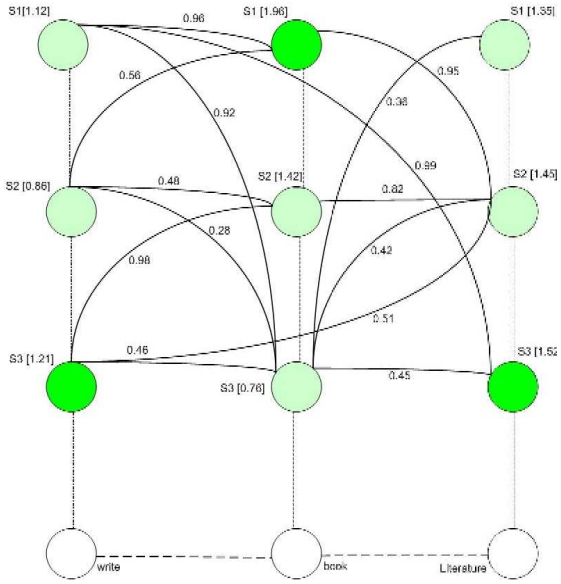

Figure 1: partial labeled graph with senses assigned to words in the sentence *He wrote a book on Literature*

2.2 Labeling the Graph Edges

Consider the sentence *He wrote a book on literature*. The $V - N_1 - P - N_2$ tuple is *wrote book on literature* and words to be disambiguated are *write, book, and literature*. The labeled graph for this *V N P N* tuple is constructed as described in Section 2.1. Figure 1 shows the labeled graph structure for this sentence. All the words in the sentence have multiple meaning. The words *write*, *book*, and *literature* have nine, ten and four WordNet senses respectively. For the simplicity of presentation, only a few sample labeled sense dependency edges are shown in the graph. If the sense dependency similarity is zero, no edge is placed. In figure 1, it can be observed that the graph-based iterative algorithm performing random walk on the context sense dependency graph assigns the highest weight to sense #3 for *write*, sense #1 for*book* and sense #3 for *literature*. The process of assigning weights to vertices are given in subsection 2.4. However, it can be observed that the differences between the first and the third sense of *write* and all the senses for the word *literature* are slight. For example, all the sense in case of literature can be interchangeably used.

2.3 Sense Similarity

The similarity between the two words is computed using the following criteria, which is motivated by the variations of original Lesk algorithms [Lesk, 1986]: (1) The longer the sequence of words that match, the more is the similarity, (2) stop words such as *as, a, the, be, is, shall* and *will* are filtered out. Prepositions are not filtered out, since our words to be sense disambiguated are associated with prepositions. If the same preposition appears in the definitions of the words that match, extra weight is assigned to the semantic similarity. If the sequence of words match along with the preposition then the similarity gets more weight, (3) Normalization of the weights is done based on the length of the definition to counter the effect of long definitions, (4) Pronouns of similar forms, such as $\{$*I, we, he, she, they, you*$\}$, are treated as a single entity, (5) Words having different derivational morphological forms such as *published and publication* are considered similar.

2.4 Labeling Process for Graph Vertices

The basic idea is motivated by the *PageRank algorithm* [Brin and Page, 1998].

1. More the number of links are connected to a vertex, more important is the vertex.
2. If the incoming link is from a very important vertex, then link itself carries more weight, accordingly the vertex receiving the link is highly weighted.

For the labeled graph, given a set of weights w_{ab} associated with the edge connecting vertices V_a and V_b,the weighted Page Rank score is determined as given in Equation 1:

$$WP(V_a) = (1-d) + d * \sum_{V_b \in In(V_a)} \frac{w_{ba}}{\sum_{V_c \in Out(V_a)} w_{bc}} WP(V_b) \quad (1)$$

In Equation 1, the $WP(V_a)$ value indicate the stationary probability of a particular sense of a word.

2.5 Iterative Algorithm for Vertex Ranking

The algorithm consists of three main steps: (1) building of the labeled dependencies graph (2) scoring of vertices using graph-based ranking algorithms (3) label assignment. The iterative algorithm for vertex ranking using Equation 1 is similar to described in [Mihalcea, 2005], with the difference in the way we find the similarity between the senses of words. The steady state weights are the relative weights of the senses. Since the graph-based algorithm ranks the weights, we have experimented with assigning multiple senses to each word.

3 Data Sparsity Reduction (DSR) Process

Words in general are polysemous. Much of the data sparsity in this context can be attributed to non-exploitation of paradigmatic relationships among words. It is rather infeasible to collect all possible combinations of training examples even from a large corpus, if paradigmatic and contextual relations are not exploited. Consider the following sentences as a sample of the corpus.

1. *He painted the wall with colour.* (va- meaning verb atachment)
2. *He paints the wall with red color.* (va)
3. *He coated the wall with colours.* (va)
4. *He will paint the room with medieval scenes.* (va)
5. *They coated the building with cracks.* (na, meaning noun attachment)
6. *I coloured the house with distemper.* (va)

The corresponding PP-attachment tags are given in parenthesis.

In the given data set, many of the words are variants of inflectional morphology, such as painted, paints, paint and a few of them are synonyms such as *colour, color* and *building, house, room*. We should exploit these observations. Moreover, if we can establish a relation between the verbs *painted* and *coated*, given the preposition *with* and *Att=va*, then possibly we can find higher probability in Equation 2, even though such an instance is not available in the training corpus.

$$P(N2 = scene|V = coated, P = with, Att = va) \quad (2)$$

The data sparsity reduction (DSR) procedure is described next.

3.1 Data Sparsity Reduction (DSR)

Use is made of *Lemmatisation, Synset replacement (paradigmatic substitution) and Inferencing based on syntagmatic context*. The updation process for *Verb Attachment* has been given below; a similar process is followed for *Noun Attachment* by substituting *N1* for *V*. **Updation process for verb Attachment:**
The following steps are applied on the *original training corpus* for each *preposition* for modeling $Pr(N_2|P, V, va)$. When the attachment is to V, N_1 is independent of N_2, and N_1 statitics is not changed.

1. **Lemmatisation and Morphing:** (a)*Lemmatisation of dependent noun:* after lemmatising if tuples become similar, add to the frequency counts of the tuples and (b)*Morph verb:* after morphing if tuples become similar, add to the frequency counts of the tuples.
2. **Synset replacement:** For each tuple in the corpus, create new tuples with weights proportional to empirical counts, for each word in the Synsets of first two senses of V and N_2. Suppose, if V has $L_1 = SynWCont(V)$ number of synonymous words and N_2 has $L_2 = SynWCont(N_2)$ number of synonymous words, then $L_1 * L_2$ number of tuples are generated, each having weight of the empirical count of original tuple. Update the appropriate frequency counts with the counts of $L_1 * L_2$ newly generated tuples.
3. **Inferencing:** The third step involves *inferencing* among the tuples in the $V - P - N_2$ syntagmatic context based on matching partly or fully, which either may generate new tuples not available in the training corpus or may increase the frequency count of the existing tuples. For example, if $V_1 - P - N_1$ and $V_2 - P - N_1$ exist as also do $V_1 - P - N_2$ and $V_2 - P - N_2$, then if $V_3 - P - N_i$ exists ($i = 1, 2$), we can infer the existence of V_3PN_j where $i \neq j$, with frequency count of V_3PN_i added appropriately.

The above three steps are applied in the given order. We have observed significant reduction DSR.

4 Unsupervised Prepositional Phrase Attachment Method

In this section, we propose an unsupervised PP-attachment method which does not require any annotated data. The proposed method directly collects training examples from the text. Our approach is based on the hypothesis that *unambiguous attachment cases of training data TEACH how to resolve the ambiguous attachment cases of the test data*. Our approach is motivated from the work of Ratnaparkhi [Ratnaparkhi, 1998] and is to some extent similar in terms of the statistical modelling and the extraction of training examples. However, we have introduced the *graph based word sense disambiguation* and *data sparsity reduction*, which is a point of difference. Another point of difference is the employing of the *training data refinement process* described in subsection 4.4.

We resolve attachment in the *ambiguous* $V - N_1 - P - N_2$ test instances using the extracted *unambiguous* $V - P - N_2$ and $N_1 - P - N_2$ cases from the training data. For example, the ambiguous *ate rice with spoon*, can be interpreted as correct unambiguous *ate with spoon* and incorrect unambiguous *rice with spoon* instances. The extracted $V - P - N_2$s are more reliable. $N_1 - P - N_2$s are not reliable- particularly the N_1PN_2s in which $P - N_2$s are the *PP-adjuncts* which can appear far away from verbs but actually are attached to verbs.

4.1 Collecting Training Data from Raw Text

We first annotate the text with part-of-speech tags using the LT POS[2]. The the noun phrases and verb phrases are chunked using our own simple chunker. After chunking we replace each chunk with their head words. Extraction heuristics are then applied to extract the unambiguous $V - P - N_2$s and $N_1 - P - N_2$ training instances given in subsection 4.2. The whole process of tagging, chunking, extraction of training examples are given in Table 1.

[2]A product of Language Technology Group (LTG), Edinburg.$http://www.ltg.ed.ac.uk/$

Table 1: The process of extracting training data from raw text

Tools.	Output
Raw Text	The professional conduct of the doctors is guided by Indian Medical association.
POS Tagger	The_DT professional_JJ conduct_NN of_IN the_DT doctors_NNS is_VBZ guided_VBN by_IN Indian_NNP Medical_NNP Association._NNP ...
Chunker	conduct_NN of_IN doctors_NNS guided_VBN by_IN Association
Extraction Heuristic	(N_1 = conduct, P= of, N_2= doctors) (v = guided, P = by, N_2 = Association)
Morphology	(N_1 = conduct, P = of, N_2 = doctor) (V = guide, P = by, N_2= association)
Synset Addition	(N_1 = conduct, P = of, N_2 = doctor) (N_1 = behavior, P= of, N_2= physician) similarly we can have 4*6= 24 combinations, and (V= guide, P= by, N_2= association) (V= direct, P= by, N_2= association) similarly we can have 9*1= 9 combinations

4.2 Heuristic Extraction of Unambiguous Training Data

The extraction heuristic exploits the idea that an attachment site of a preposition is usually within a few words to the left of the preposition. The heuristic has the following parameters:

Window size S: This is the maximum distance in words between a preposition P and N_1, V or N_2. We use *W = 4* in our experiments. We extract:

1. $V - P - N_2$, if the parsed segments satisfy:
   - P is a preposition
   - V is not a form of the verb *to be*
   - V is the first verb that occurs within W words to the left of P
   - No noun occurs between V and P
   - N_2 is the first noun that occurs within W words to the right of P
   - No verb occurs between P and N_2
2. $N_1 - P - N_2$, if the parsed segments satisfy:
   - P is a preposition
   - N is the first noun that occurs within W words to the left of P
   - No verb occurs within W words to the left of P. If it appears, it must be ensured that a preposition, subordinating conjunction or a Wh-type conjunction appears between N and the new verb seen
   - N_2 is the first noun that occurs within W words to the right of P
   - No verb occurs between P and N_2

Since the unambiguous instances $V - P - N_2$s and $N_1 - P - N_2$s are extracted using heuristics, these- particularly $N_1 - P - N_2$s- are not always correct, and hence call for refinement.

4.3 Refinement of Training Data

We filter out the incorrect $V - P - N_2$ and $N_1 - P - N_2$ instances by applying the graph based WSD algorithm discussed in 2.1. The features considered are one word to the right of N_2 and one word to left of V or N_1. Only the nouns and verbs are disambiguated. Three strategies are used to refine the data:

1. Set of heuristics for reliable unambiguous $N_1 - P - N_2$s. These are based on syntactic heuristics which pick almost reliable NPNs. For example, (1) $N_1 - P - N_2$ as *subject*: like *tube through doorway* in the sentence *The tube through the doorway disturbs the people* and (2) $N_1 - P - N_2$ as predicate in the B part of a sentence of the form *A <form of 'be'> B*: as in *item in program* in the sentence *It is an important item in the program*.
2. Tuples after step 1 are further refined using strong conditions. We use WordNet to find semantic properties of words such as *place, time, group* etc.
3. Finally slightly weaker conditions are applied through limited statistical inferencing to give a set of highly correct $V - P - N_2$ and $N_1 - P - N_2$ tuples.

4.4 Training Method

Our goal is to resolve the ambiguous PP-attachment instances using the *learnt* knowledge of unambiguous PP-attachment. The ambiguous tuple are of the form *V N1 P N2*. The unambiguous training tuples are of the form $V - P - N_2$ and $N_1 - P - N_2$. We define our classier as in Equation 3.

$$ATT(V, N_1, P, N_2) = \arg \max_{a \in \{N,V\}} Pr(V, N_1, P, N_2, a) \tag{3}$$

We can factor $Pr(V, N_1, P, N_2, a)$ as follows:

$$Pr(V, N_1, P, N_2, a) = \begin{cases} Pr(V)Pr(N_1)Pr(a|V, N_1)Pr(P|a, V, N_1) \\ P(N_2|P, a, V, N_1) \end{cases} \tag{4}$$

The factors $Pr(N_1)$ and $Pr(V)$ are independent of the attachment a and need not be computed. The estimation of $Pr(a|V, N_1)$, $Pr(P|a, V, N_1)$, and $Pr(N_2|P, a, V, N_1)$ is difficult, because in the training data both N_1 and V do not occur together. For this reason, these factors are computed using the approximation in Equation 5:

$$\begin{aligned} Pr(a = N_1|V, N_1) &\approx \tfrac{Pr(a=N_1|N_1)}{Z(V,N_1)} \\ Pr(a = V|V, N_1) &\approx \tfrac{Pr(a=V|V)}{Z(V,N_1)} \end{aligned} \tag{5}$$

where, $Z(V, N_1) = Pr(a = N_1|N_1) + Pr(a = V|V)$. Similarly, we approximate $Pr(P|a, V, N_1)$ and $Pr(N_2|P, a, V, N_1)$ as given in Equations 6 and 7 respectively. The reasons for these approximations are to avoid using counts of (V, N_1) together, since they are never seen together in the extracted data.

$$\begin{aligned} Pr(P|a = N_1, V, N_1) &\approx Pr(P|a = N_1, N_1) \\ Pr(P|a = V, V, N_1) &\approx Pr(P|a = V, V) \end{aligned} \tag{6}$$

$$\begin{aligned} Pr(N_2|P, a = N_1, V, N_1) &\approx Pr(N_2|P, a = N_1, N_1) \\ Pr(N_2|P, a = V, V, N_1) &\approx Pr(N_2|P, a = V, V) \end{aligned} \tag{7}$$

The approximated probabilities are computed from the training data as in [Medimi and Bhattacharyya, 2004]. We used a variant of backed-off technique in order to smooth the probability computation.

5 Experiments, Results and Analysis

Training Data: We used Brown corpus for collecting the unambiguous training examples. The corpus size is 6MB, consisting of 51763 sentences, and nearly 1 million 27 thousand words. The most frequent prepositions are *in, to, for, with, on, at, from*. The preposition *of* which is highly biased towards noun attachment is not considered. The extracted unambiguous distinct $n_1 - p - n_2$ and $v - p - n_2$ tuples number 54030 and 22362 respectively.

Testing Data: For testing, we used Penn Treebank Wall Street Journal data by Ratnaparkhi [Ratnaparkhi *et al.*, 1994], which is a standard benchmark data for PP attachment used by many groups [Ratnaparkhi *et al.*, 1994; Collins and Brooks, 1995; Stetina and Nagao, 1997].

Baseline: We consider the unsupervised approach by Ratnaparkhi [Ratnaparkhi, 1998] as the Baseline system for our performance evaluation. We name it as *Base-RP*. Further, we tested the performance of our system on the extracted unambiguous samples. We name this process as *Base-MS*.

We name our proposed system *A Flexible Unsupervised PP-attachment* or in short *FlxUppAttch*. We experimented on the performance of *FlxUppAttch* in two stages: (i) *DSR* without *WSD* (we call it *DSR-wo-WSD*) and (ii) *DSR* with *WSD* (we call it *DSR-with-WSD*). The stages and their names appear in Table 2.

Table 2: The naming of *FlxUppAttch* systems utilizing different DSR stages and the graph based WSD

	Stages of Data Sparsity Reduction			
	Morphing	**Inferencing**	**Synset**	**Synset & Inferencing**
DSR-wo-WSD	*Morph*	*Infer*	*WnSyn*	*Syn-Inf*
DSR-with-WSD	*MorphWS*	*InferWS*	*WnSynWS*	*Syn-InfWS*

Further, since our WSD method provides ranks to all the senses of each word, we experimented with different schemes of sense assignment

1. **GwsRnk1**- assign the first ranked sense
2. **GwsRnk2**- first two highest ranked senses
3. **GwsRnk3**- first three highest ranked senses
4. **GwsRnk3-C1**- first sense always, 50% times the random assignment of the second sense and 30% times the random assignment of the third sense
5. **GwsRnk3-C2**- 50% times the first sense, 30% times the second sense, and 20% times the third sense

With different senses being assigned, we observed the performance of our system with respect to different stages of DSR process particulrly after synsets and after synsets and inferencing. This performance comparision is given in Figure 2. In case of *GwsRnk3*, the precision is low. This is because, though the coverages of the tuples increases due to , it introduces noise through wrong lexical entries, which has a net negetive effect on the precisison. The best performing combination is (*GwsRnk3-C2)*), this may be due to the fact

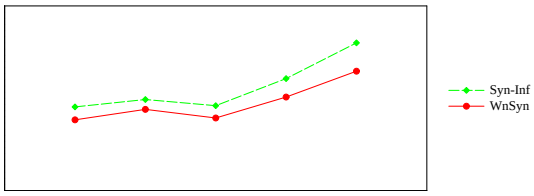

Figure 2: Performance summary of FlxUppAttch DSR-with-WSD with combination of senses assigned to words

that, it makes a trade-off between coverage and precision. In case of (*GwsRnk3-C1)*), though it increases the precision, it decreases the coverage proportionately. It may also be observed that inferencing consistently increases the precision of the system.

We also compared the the performance of our system with baseline using WSD and applying the DSR in stages. Figure 3 shows the performance variation.

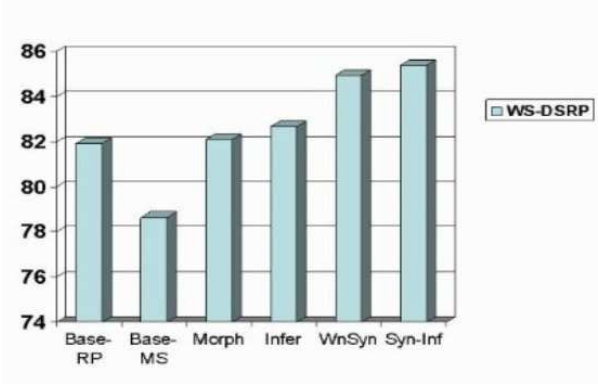

Figure 3: Comparison of Baselines with *FlxUppAttch*(*GwsRnk3-C2*) at different stages of DSR

The performance of our system after morphing is better than performance of Ratnaparkhi [Ratnaparkhi, 1998], *i.e Base-RP*. This gives an indication of the better accuracy of extraction heuristics. The comparative performance of our best performing system with the state-of-the-art systems are shown in Table 3

6 Conclusion

We presented an unsupervised method for PP-attachment which compares favourably with the existing state of the art approaches. The method makes use of lexical semantics and inferencing through the use of WordNet. We employ WSD in a limited way and make use of the unambiguous PP-attachments to learn the resolution of PP-Attachment in case of the ambiguous $V - N_1 - P - N_2$ tuples. Since the starting point is raw corpora, the method is usable even when annotation is not available. Clearly the efficacy of the method depends on the richness of the presence of $V - P - N$ and $N - P - N$ tuples. Obvius future work consists in refining

Table 3: Comparison of FlxUppAttch (*GwsRnk3-C2*) with state-of-the-art-systems

	PP-attachment systems	**Precision(%)**
Human without context		
1	Ratnaparkhi [Ratnaparkhi *et al.*, 1994]	88.2
2	Mitchell 2003	78.3
Use of WordNet back off		
3	Stetina and Nagao [Stetina and Nagao, 1997]	88.1
4	Li and Abe 1998	85.2
5	**FlxUppAttch(GwsRnk3-C2)**	**85.4**
Use of thesaurus back off		
6	Pantel and Lin [Pantel and Lin, 2000]	84.3
7	McLauchlan 2004	85.0
8	Zhao and Lin 2004	86.5

WSD and thereby improving the performance of attachment, and also dealing with the more difficult case of $N - P - N$ tuples.

References

[Brin and Page, 1998] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer networks and ISDN Systems,*, pages 30(1–7), 1998.

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

[Donald Hindle and Rooth, 1993] Donald Donald Hindle and Mats Rooth. Structural ambiguity and lexical relations. In *Computational Linguistics*, pages 19(1), 103–120, 1993.

[Lesk, 1986] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *proceedings of SIGDOC'86*, 1986.

[Medimi and Bhattacharyya, 2004] Srinivas Medimi and Pushpak Bhattacharyya. Unsupervised pp attachment disambiguation using semantics. In *Third International Conference on Natural Language Processing (ICON 2004)*, Hyderabad, INDIA, 19-22 December 2004.

[Mihalcea, 2005] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *proceedings of HLT/EMNLP*, Vancouver, B.C., Canada, October 6-8 2005.

[Pantel and Lin, 2000] P. Pantel and D. Lin. An unsupervised approach to prepositional phrase attachment using contextual similar words. In *Proceedings of Association for Computational Linguistics (ACL-00)*, 2000.

[Ratnaparkhi *et al.*, 1994] A. Ratnaparkhi, J. Reynar, and S. Roukos. Maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Workshop on Human Language Technology*, 1994.

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

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