# An Unsupervised Approach to Prepositional Phrase Attachment using Contextually Similar Words

**Patrick Pantel** and **Dekang Lin**
Department of Computing Science
University of Alberta[1]
Edmonton, Alberta T6G 2H1 Canada
{ppantel, lindek}@cs.ualberta.ca

## Abstract

Prepositional phrase attachment is a common source of ambiguity in natural language processing. We present an unsupervised corpus-based approach to prepositional phrase attachment that achieves similar performance to supervised methods. Unlike previous unsupervised approaches in which training data is obtained by heuristic extraction of unambiguous examples from a corpus, we use an iterative process to extract training data from an automatically parsed corpus. Attachment decisions are made using a linear combination of features and low frequency events are approximated using contextually similar words.

## Introduction

Prepositional phrase attachment is a common source of ambiguity in natural language processing. The goal is to determine the attachment site of a prepositional phrase in a sentence. Consider the following examples:

1. Mary ate the salad with a fork.
2. Mary ate the salad with croutons.

In both cases, the task is to decide whether the prepositional phrase headed by the preposition *with* attaches to the noun phrase (NP) headed by *salad* or the verb phrase (VP) headed by *ate*. In the first sentence, *with* attaches to the VP since Mary is using a fork to eat her salad. In sentence 2, *with* attaches to the NP since it is the salad that contains croutons.

Formally, prepositional phrase attachment is simplified to the following classification task. Given a 4-tuple of the form ($V$, $N_1$, $P$, $N_2$), where $V$ is the head verb, $N_1$ is the head noun of the object of $V$, $P$ is a preposition, and $N_2$ is the head noun of the prepositional complement, the goal is to classify as either adverbial attachment (attaching to $V$) or adjectival attachment (attaching to $N_1$). For example, the 4-tuple (*eat*, *salad*, *with*, *fork*) has target classification $V$.

In this paper, we present an unsupervised corpus-based approach to prepositional phrase attachment that outperforms previous unsupervised techniques and approaches the performance of supervised methods. Unlike previous unsupervised approaches in which training data is obtained by heuristic extraction of unambiguous examples from a corpus, we use an iterative process to extract training data from an automatically parsed corpus. The attachment decision for a 4-tuple ($V$, $N_1$, $P$, $N_2$) is made as follows. First, we replace $V$ and $N_2$ by their contextually similar words and compute the average adverbial attachment score. Similarly, the average adjectival attachment score is computed by replacing $N_1$ and $N_2$ by their contextually similar words. Attachment scores are obtained using a linear combination of features of the 4-tuple. Finally, we combine the average attachment scores with the attachment score of $N_2$ attaching to the original $V$ and the attachment score of $N_2$ attaching to the original $N_1$. The proposed classification represents the attachment site that scored highest.

## 1 Previous Work

Altmann and Steedman (1988) showed that current discourse context is often required for

[1]This research was conducted at the University of Manitoba.

disambiguating attachments. Recent work shows that it is generally sufficient to utilize lexical information (Brill and Resnik, 1994; Collins and Brooks, 1995; Hindle and Rooth, 1993; Ratnaparkhi et al., 1994).

One of the earliest corpus-based approaches to prepositional phrase attachment used lexical preference by computing co-occurrence frequencies (lexical associations) of verbs and nouns with prepositions (Hindle and Rooth, 1993). Training data was obtained by extracting all phrases of the form ($V$, $N_1$, $P$, $N_2$) from a large parsed corpus.

Supervised methods later improved attachment accuracy. Ratnaparkhi et al. (1994) used a maximum entropy model considering only lexical information from within the verb phrase (ignoring $N_2$). They experimented with both word features and word class features, their combination yielding 81.6% attachment accuracy.

Later, 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.

A non-statistical supervised approach by Brill and Resnik (1994) yielded 81.8% accuracy using a transformation-based approach (Brill, 1995) and incorporating word-class information. They report that the top 20 transformations learned involved specific prepositions supporting Collins and Brooks' claim that the preposition is the most important lexical item for resolving the attachment ambiguity.

The state of the art is a supervised algorithm that employs a semantically tagged corpus (Stetina and Nagao, 1997). Each word in a labelled corpus is sense-tagged using an unsupervised word-sense disambiguation algorithm with WordNet (Miller, 1990). 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 current unsupervised state of the art achieves 81.9% attachment accuracy (Ratnaparkhi, 1998). Using an extraction heuristic, unambiguous prepositional phrase attachments of the form ($V$, $P$, $N_2$) and ($N_1$, $P$, $N_2$) are extracted from a large corpus. Co-occurrence frequencies are then used to disambiguate examples with ambiguous attachments.

| | |
|---|---|
| **eat**: | |
| object: | almond 1, apple 25, bean 5, beam 1, binge 1, bread 13, cake 17, cheese 8, dish 14, disorder 20, egg 31, grape 12, grub 2, hay 3, junk 1, meat 70, poultry 3, rabbit 4, soup 5, sandwich 18, pasta 7, vegetable 35, ... |
| subject: | adult 3, animal 8, beetle 1, cat 3, child 41, decrease 1, dog 24, family 29, guest 7, kid 22, patient 7, refugee 2, rider 1, Russian 1, shark 2, something 19, We 239, wolf 5, ... |
| **salad**: | |
| adj-modifier: | assorted 1, crisp 4, fresh 13, good 3, grilled 5, leftover 3, mixed 4, olive 3, prepared 3, side 4, small 6, special 5, vegetable 3, ... |
| object-of: | add 3, consume 1, dress 1, grow 1, harvest 2, have 20, like 5, love 1, mix 1, pick 1, place 3, prepare 4, return 3, rinse 1, season 1, serve 8, sprinkle 1, taste 1, test 1, Toss 8, try 3, ... |

*Figure 1.* Excepts of entries in the collocation database for *eat* and *salad*.

*Table 1.* The top 20 most similar words of *eat* and *salad* as given by (Lin, 1998b).

| WORD | SIMILAR WORDS (WITH SIMILARITY SCORE) |
|---|---|
| EAT | cook 0.127, drink 0.108, consume 0.101, feed 0.094, taste 0.093, like 0.092, serve 0.089, bake 0.087, sleep 0.086, pick 0.085, fry 0.084, freeze 0.081, enjoy 0.079, smoke 0.078, harvest 0.076, love 0.076, chop 0.074, sprinkle 0.072, Toss 0.072, chew 0.072 |
| SALAD | soup 0.172, sandwich 0.169, sauce 0.152, pasta 0.149, dish 0.135, vegetable 0.135, cheese 0.132, dessert 0.13, entree 0.121, bread 0.116, meat 0.116, chicken 0.115, pizza 0.114, rice 0.112, seafood 0.11, dressing 0.109, cake 0.107, steak 0.105, noodle 0.105, bean 0.102 |

## 2 Resources

The input to our algorithm includes a collocation database and a corpus-based thesaurus, both available on the Internet[2]. Below, we briefly describe these resources.

### 2.1 Collocation database

Given a word $w$ in a dependency relationship (such as *subject* or *object*), the collocation database is used to retrieve the words that occurred in that relationship with $w$, in a large corpus, along with their frequencies (Lin, 1998a). Figure 1 shows excerpts of the entries in

[2]Available at www.cs.ualberta.ca/~lindek/demos.htm.

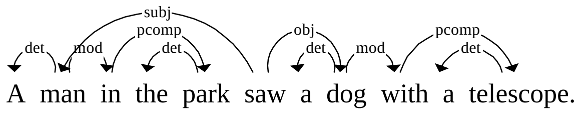
the collocation database for the words *eat* and *salad*. The database contains a total of 11 million unique dependency relationships.

## 2.2 Corpus-based thesaurus

Using the collocation database, Lin (1998b) used an unsupervised method to construct a corpus-based thesaurus consisting of 11839 nouns, 3639 verbs and 5658 adjectives/adverbs. Given a word *w*, the thesaurus returns a set of similar words of *w* along with their similarity to *w*. For example, the 20 most similar words of *eat* and *salad* are shown in Table 1.

## 3 Training Data Extraction

We parsed a 125-million word newspaper corpus with *Minipar*[3], a descendent of Principar (Lin, 1994). Minipar outputs dependency trees (Lin, 1999) from the input sentences. For example, the following sentence is decomposed into a dependency tree:



Occasionally, the parser generates incorrect dependency trees. For example, in the above sentence, the prepositional phrase headed by *with* should attach to *saw* (as opposed to *dog*). Two separate sets of training data were then extracted from this corpus. Below, we briefly describe how we obtained these data sets.

## 3.1 Ambiguous Data Set

For each input sentence, Minipar outputs a single dependency tree. For a sentence containing one or more prepositions, we use a program to detect any alternative prepositional attachment sites. For example, in the above sentence, the program would detect that *with* could attach to *saw*. Using an iterative algorithm, we initially create a table of co-occurrence frequencies for 3-tuples of the form ($V$, $P$, $N_2$) and ($N_1$, $P$, $N_2$). For each $k$ possible attachment site of a preposition $P$, we increment the frequency of the corresponding 3-tuple by $1/k$. For example, Table 2 shows the initial co-occurrence frequency table for the corresponding 3-tuples of the above sentence.

[3]Available at www.cs.ualberta.ca/~lindek/minipar.htm.

*Table 2.* Initial co-occurrence frequency table entries for *A man in the park saw a dog with a telescope.*

| V OR $N_1$ | P | $N_2$ | FREQUENCY |
|---|---|---|---|
| man | in | park | 1.0 |
| saw | with | telescope | 0.5 |
| dog | with | telescope | 0.5 |

*Table 3.* Co-occurrence frequency table entries for *A man in the park saw a dog with a telescope* after one iteration.

| V OR $N_1$ | P | $N_2$ | FREQUENCY |
|---|---|---|---|
| man | in | park | 1.0 |
| saw | with | telescope | 0.913 |
| dog | with | telescope | 0.087 |

In the following iterations of the algorithm, we update the frequency table as follows. For each $k$ possible attachment site of a preposition $P$, we refine its attachment score using the formulas described in Section 4: $VScore(V_k, P_k, N_{2k})$ and $NScore(N_{1k}, P_k, N_{2k})$. For any tuple ($W_k$, $P_k$, $N_{2k}$), where $W_k$ is either $V_k$ or $N_{2k}$, we update its frequency as:

$$fr(W_k, P_k, N_{2k}) = \frac{Score(W_k, P_k, N_{2k})}{\sum_{i=1}^{k} Score(W_i, P_i, N_{2i})}$$

where $Score(W_k, P_k, N_{2k}) = VScore(W_k, P_k, N_{2k})$ if $W_k = V_k$; otherwise $Score(W_k, P_k, N_{2k}) = NScore(W_k, P_k, N_{2k})$.

Suppose that after the initial frequency table is set *NScore*(*man, in, park*) = 1.23, *VScore*(*saw, with, telescope*) = 3.65, and *NScore*(*dog, with, telescope*) = 0.35. Then, the updated co-occurrence frequencies for (*man*, *in*, *park*) and (*saw*, *with*, *telescope*) are:

$$fr_{(man,\ in,\ park)} = \frac{1.23}{1.23} = 1.0$$
$$fr_{(saw,\ \text{with},\ telescope)} = \frac{3.65}{3.65+0.35} = 0.913$$

Table 3 shows the updated frequency table after the first iteration of the algorithm. The resulting database contained 8,900,000 triples.

## 3.2 Unambiguous Data Set

As in (Ratnaparkhi, 1998), we constructed a training data set consisting of only unambiguous attachments of the form ($V$, $P$, $N_2$) and ($N_1$, $P$, $N_2$). We only extract a 3-tuple from a sentence when our program finds no alternative attachment site for its preposition. Each extracted 3-tuple is assigned a frequency count of 1. For example, in the previous sentence, (*man*, *in*, *park*) is extracted since it contains only one attachment site; (*dog*, *with*, *telescope*) is not extracted since *with* has an alternative attachment site. The resulting database contained 4,400,000 triples.

## 4 Classification Model

Roth (1998) presented a unified framework for natural language disambiguation tasks. Essentially, several language learning algorithms (e.g. naïve Bayes estimation, back-off estimation, transformation-based learning) were successfully cast as learning linear separators in their feature space. Roth modelled prepositional phrase attachment as linear combinations of features. The features consisted of all 15 possible sub-sequences of the 4-tuple ($V$, $N_1$, $P$, $N_2$) shown in Table 4. The asterix (*) in features represent wildcards.

Roth used supervised learning to adjust the weights of the features. In our experiments, we only considered features that contained $P$ since the preposition is the most important lexical item (Collins and Brooks, 1995). Furthermore, we omitted features that included both $V$ and $N_1$ since their co-occurrence is independent of the attachment decision. The resulting subset of features considered in our system is shown in bold in Table 4 (equivalent to assigning a weight of 0 or 1 to each feature).

Let |*head*, *rel*, *mod*| represent the frequency, obtained from the training data, of the *head* occurring in the given relationship *rel* with the *modifier*. We then assign a score to each feature as follows:

1. $(*, *, P, *) = log(|*, P, *| / |*, *, *|)$
2. $(V, *, P, N_2) = log(|V, P, N_2| / |*, *, *|)$
3. $(*, N_1, P, N_2) = log(|N_1, P, N_2| / |*, *, *|)$
4. $(V, *, P, *) = log(|V, P, *| / |V, *, *|)$
5. $(*, N_1, P, *) = log(|N_1, P, *| / |N_1, *, *|)$
6. $(*, *, P, N_2) = log(|*, P, N_2| / |*, *, N_2|)$

1, 2, and 3 are the prior probabilities of $P$, $V$ $P$ $N_2$, and $N_1$ $P$ $N_2$ respectively. 4, 5, and 6 represent conditional probabilities P($V$, $P$ | $V$), P($N_1$, $P$ | $N_1$), and P($P$ $N_2$ | $N_2$) respectively.

*Table 4*. The 15 features for prepositional phrase attachment.

| FEATURES | | |
|---|---|---|
| ($V$, *, *, *) | **($V$, *, $P$, *)** | (*, $N_1$, *, $N_2$) |
| ($V$, $N_1$, *, *) | ($V$, *, *, $N_2$) | **(*, $N_1$, $P$, $N_2$)** |
| ($V$, $N_1$, $P$, *) | **($V$, *, $P$, $N_2$)** | **(*, *, $P$, *)** |
| ($V$, $N_1$, *, $N_2$) | (*, $N_1$, *, *) | (*, *, *, $N_2$) |
| ($V$, $N_1$, $P$, $N_2$) | **(*, $N_1$, $P$, *)** | **(*, *, $P$, $N_2$)** |

We estimate the adverbial and adjectival attachment scores, *VScore*($V$, $P$, $N_2$) and *NScore*($N_1$, $P$, $N_2$), as a linear combination of these features:

$$VScore(V, P, N_2) = (*, *, P, *) + (V, *, P, N_2) + (V, *, P, *) + (*, *, P, N_2)$$

$$NScore(N_1, P, N_2) = (*, *, P, *) + (*, N_1, P, N_2) + (*, N_1, P, *) + (*, *, P, N_2)$$

For example, the attachment scores for (*eat*, *salad*, *with*, *fork*) are *VScore*(*eat*, *with*, *fork*) = -3.47 and *NScore*(*salad*, *with*, *fork*) = -4.77. The model correctly assigns a higher score to the adverbial attachment.

## 5 Contextually Similar Words

The contextually similar words of a word *w* are words similar to the intended meaning of *w* in its context. Below, we describe an algorithm for constructing contextually similar words and we present a method for approximating the attachment scores using these words.

### 5.1 Algorithm

For our purposes, a context of *w* is simply a dependency relationship involving *w*. For example, a dependency relationship for *saw* in the example sentence of Section 3 is *saw*:*obj*:*dog*. Figure 2 gives the data flow diagram for our algorithm for constructing the contextually similar words of *w*. We retrieve from the collocation database the words that occurred in the same dependency relationship as *w*. We refer to this set of words as the **cohort** of *w* for the dependency relationship. Consider the words *eat* and *salad* in the context *eat salad*. The cohort of *eat* consists of verbs that appeared with object *salad* in Figure 1 (e.g. add, consume, cover, …) and the cohort of *salad* consists of nouns that appeared as object of *eat* in Figure 1 (e.g. almond, apple, bean, …).

Intersecting the set of similar words and the cohort then forms the set of contextually similar words of *w*. For example, Table 5 shows the contextually similar words of *eat* and *salad* in the context *eat salad* and the contextually similar words of *fork* in the contexts *eat with fork* and *salad with fork*. The words in the first row are retrieved by intersecting the similar words of *eat* in Table 1 with the cohort of *eat* while the second row represents the intersection of the similar words of *salad* in Table 1 and the cohort of *salad*. The third and fourth rows are determined in a similar manner. In the nonsensical context *salad with fork* (in row 4), no contextually similar words are found.

While previous word sense disambiguation algorithms rely on a lexicon to provide sense inventories of words, the contextually similar words provide a way of distinguishing between different senses of words without committing to any particular sense inventory.

## 5.2 Attachment Approximation

Often, sparse data reduces our confidence in the attachment scores of Section 4. Using contextually similar words, we can approximate these scores. Given the tuple ($V$, $N_1$, $P$, $N_2$), adverbial attachments are approximated as follows. We first construct a list $CS_V$ containing the contextually similar words of $V$ in context $V$:$obj$:$N_1$ and a list $CS_{N_2\mathrm{V}}$ containing the contextually similar words of $N_2$ in context $V$:$P$:$N_2$ (i.e. assuming adverbial attachment). For each verb $v$ in $CS_V$, we compute $VScore(v, P, N_2)$ and set $S_V$ as the average of the largest $k$ of these scores. Similarly, for each noun $n$ in $CS_{N_2\mathrm{V}}$, we compute $VScore(V, P, n)$ and set $S_{N_2\mathrm{V}}$ as the average of the largest $k$ of these scores. Then, the approximated adverbial attachment score, *Vscore'*, is:

$$VScore'(V, P, N_2) = max(S_V, S_{N_2V})$$

We approximate the adjectival attachment score in a similar way. First, we construct a list $CS_{N_1}$ containing the contextually similar words of $N_1$ in context $V$:$obj$:$N_1$ and a list $CS_{N_2N_1}$ containing the contextually similar words of $N_2$ in context $N_1$:$P$:$N_2$ (i.e. assuming adjectival attachment). Now, we compute $S_{N_1}$ as the average of the largest $k$ of $NScore(n, P, N_2)$ for each noun $n$ in $CS_{N_1}$ and $S_{N_2N_1}$ as the average of the largest $k$ of $NScore(N_1, P, n)$ for each noun $n$ in $CS_{N_2N_1}$. Then, the approximated adjectival attachment score, *NScore'*, is:

$$NScore'(N_1, P, N_2) = max(S_{N_1}, S_{N_2N_1})$$

For example, suppose we wish to approximate the attachment score for the 4-tuple (*eat*, *salad*, *with*, *fork*). First, we retrieve the contextually similar words of *eat* and *salad* in context *eat salad*, and the contextually similar words of *fork* in contexts *eat with fork* and *salad with fork* as shown in Table 5. Let $k$ = 2. Table 6 shows the calculation of $S_V$ and $S_{N_2\mathrm{V}}$ while the calculation of $S_{N_1}$ and $S_{N_2N_1}$ is shown in Table 7. Only the

*Table 5*. Contextually similar words of *eat* and *salad*.

| WORD | CONTEXT | CONTEXTUALLY SIMILAR WORDS |
|---|---|---|
| EAT | *eat salad* | consume, taste, like, serve, pick, harvest, love, sprinkle, Toss, … |
| SALAD | *eat salad* | soup, sandwich, pasta, dish, cheese, vegetable, bread, meat, cake, bean, … |
| FORK | *eat with fork* | spoon, knife, finger |
| FORK | *salad with fork* | --- |

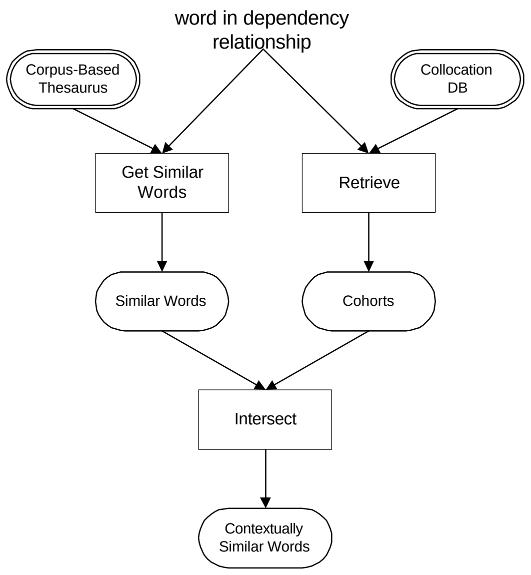


*Figure 2*. Data flow diagram for identifying the contextually similar words of a word in a dependency relationship.

top $k$ = 2 scores are shown in these tables. We have:

$$VScore' (eat, with, fork) = max(S_V, S_{N2V}) = -2.92$$

$$NScore' (salad, with, fork) = max(S_{N1}, S_{N2N1}) = -4.87$$

Hence, the approximation correctly prefers the adverbial attachment to the adjectival attachment.

## 6 Attachment Algorithm

Figure 3 describes the prepositional phrase attachment algorithm. As in previous approaches, examples with $P$ = *of* are always classified as adjectival attachments.

Suppose we wish to approximate the attachment score for the 4-tuple (*eat*, *salad*, *with*, *fork*). From the previous section, Step 1 returns $average_V$ = -2.92 and $average_{N1}$ = -4.87.

From Section 4, Step 2 gives $a_V$ = -3.47 and $a_{N1}$ = -4.77. In our training data, $f_V$ = 2.97 and $f_{N1}$ = 0, thus Step 3 gives $f$ = 0.914. In Step 4, we compute:

$S(V)$ = -3.42 and

$S(N_1)$ = -4.78

Since $S(V) > S(N_1)$, the algorithm correctly classifies this example as an adverbial attachment.

Given the 4-tuple (*eat*, *salad*, *with*, *croutons*), the algorithm returns $S(V)$ = -4.31 and $S(N_1)$ = -3.88. Hence, the algorithm correctly attaches the prepositional phrase to the noun *salad*.

```
Input:  A 4-tuple ($V$, $N_1$, $P$, $N_2$)
Step 1: Using the contextually similar words algorithm
        and the formulas from Section 5.2 compute:
            $average_V = VScore'(V, P, N_2)$
            $average_{N1} = NScore'(N_1, P, N_2)$
Step 2: Compute the adverbial attachment score, $a_v$,
        and the adjectival attachment score, $a_{n1}$:
            $a_V = VScore(V, P, N_2)$
            $a_{N1} = NScore(N_1, P, N_2)$
Step 3: Retrieve from the training data set the
        frequency of the 3-tuples ($V, P, N_2$) and
        ($N_1, P, N_2$) → $f_V$ and $f_{N1}$, respectively.
        Let $f = (f_V + f_{N1} + 0.2) / (f_V + f_{N1} + 0.5)$
Step 4: Combine the scores of Steps 1-3 to obtain the
        final attachment scores:
            $S(V) = fa_v + (1 - f)average_v$
            $S(N_1) = fa_{n1} + (1 - f)average_{n1}$
Output: The attachment decision: $N_1$ if $S(N_1) > S(V)$ or
        $P$ = of; $V$ otherwise.
```

*Figure 3.* The prepositional phrase attachment algorithm.

*Table 6.* Calculation of $S_V$ and $S_{N2V}$ for (*eat*, *salad*, *with*, *fork*).

| 4-TUPLE | VSCORE |
|---|---|
| (mix, salad, with, fork) | -2.60 |
| (sprinkle, salad, with, fork) | -3.24 |
| $\boldsymbol{S_V}$ | **-2.92** |
| (eat, salad, with, spoon) | -3.06 |
| (eat, salad, with, finger) | -3.50 |
| $\boldsymbol{S_{N2V}}$ | **-3.28** |

*Table 7.* Calculation of $S_{N1}$ and $S_{N2N1}$ for (eat, salad, with, fork).

| 4-TUPLE | NSCORE |
|---|---|
| (eat, pasta, with, fork) | -4.71 |
| (eat, cake, with, fork) | -5.02 |
| $\boldsymbol{S_{N1}}$ | **-4.87** |
| --- | n/a |
| --- | n/a |
| $\boldsymbol{S_{N2N1}}$ | **n/a** |

## 7 Experimental Results

In this section, we describe our test data and the baseline for our experiments. Finally, we present our results.

### 7.1 Test Data

The test data consists of 3097 examples derived from the manually annotated attachments in the Penn Treebank Wall Street Journal data (Ratnaparkhi et al., 1994)[4]. Each line in the test data consists of a 4-tuple and a target classification: $V$ $N_1$ $P$ $N_2$ *target*.

The data set contains several erroneous tuples and attachments. For instance, 133 examples contain the word *the* as $N_1$ or $N_2$. There are also improbable attachments such as (*sing*, *birthday*, *to*, *you*) with the target attachment *birthday*.

[4]Available at ftp.cis.upenn.edu/pub/adwait/PPattachData.

## 7.2 Baseline

Choosing the most common attachment site, $N_1$, yields an accuracy of 58.96%. However, we achieve 70.39% accuracy by classifying each occurrence of $P = of$ as $N_1$, and $V$ otherwise. Human accuracy, given the full context of a sentence, is 93.2% and drops to 88.2% when given only tuples of the form ($V$, $N_1$, $P$, $N_2$) (Ratnaparkhi et al., 1994). Assuming that human accuracy is the upper bound for automatic methods, we expect our accuracy to be bounded above by 88.2% and below by 70.39%.

## 7.3 Results

We used the 3097-example testing corpus described in Section 7.1. Table 8 presents the precision and recall of our algorithm and Table 9 presents a performance comparison between our system and previous supervised and unsupervised approaches using the same test data. We describe the different classifiers below:

| | |
|---|---|
| $cl_{base}$: | the baseline described in Section 7.2 |
| $cl_{R1}$: | uses a maximum entropy model (Ratnaparkhi et al., 1994) |
| $cl_{BR}$[5]: | uses transformation-based learning (Brill and Resnik, 1994) |
| $cl_{CB}$: | uses a backed-off model (Collins and Brooks, 1995) |
| $cl_{SN}$: | induces a decision tree with a sense-tagged corpus, using a semantic dictionary (Stetina and Nagao, 1997) |
| $cl_{HR}$[6]: | uses lexical preference (Hindle and Rooth, 1993) |
| $cl_{R2}$: | uses a heuristic extraction of unambiguous attachments (Ratnaparkhi, 1998) |
| $cl_{PL}$: | uses the algorithm described in this paper |

Our classifier outperforms all previous unsupervised techniques and approaches the performance of supervised algorithm.

We reconstructed the two earlier unsupervised classifiers $cl_{HR}$ and $cl_{R2}$. Table 10 presents the accuracy of our reconstructed classifiers. The originally reported accuracy for $cl_{R2}$ is within the 95% confidence interval of our reconstruction. Our reconstruction of $cl_{HR}$ achieved slightly higher accuracy than the original report.

[5]The accuracy is reported in (Collins and Brooks, 1995).

[6]The accuracy was obtained on a smaller test set but, from the same source as our test data.

*Table 8*. Precision and recall for attachment sites $V$ and $N_1$.

| CLASS | ACTUAL | CORRECT | INCORRECT | PRECISION | RECALL |
|---|---|---|---|---|---|
| $V$ | 1203 | 994 | 209 | 82.63% | 78.21% |
| $N_1$ | 1894 | 1617 | 277 | 84.31% | 88.55% |

*Table 9*. Performance comparison with other approaches.

| METHOD | LEARNING | ACCURACY |
|---|---|---|
| $CL_{BASE}$ | --- | 70.39% |
| $CL_{R1}$ | supervised | 81.6% |
| $CL_{BR}$ | supervised | 81.9% |
| $CL_{CB}$ | supervised | 84.5% |
| $CL_{SN}$ | supervised | 88.1% |
| $CL_{HR}$ | unsupervised | 75.8% |
| $CL_{R2}$ | unsupervised | 81.91% |
| $CL_{PL}$ | unsupervised | 84.31% |

*Table 10*. Accuracy of our reconstruction of (Hindle & Rooth, 1993) and (Ratnaparkhi, 1998).

| METHOD | ORIGINAL REPORTED ACCURACY | RECONSTRUCTED SYSTEM ACCURACY (95% CONF) |
|---|---|---|
| $CL_{HR}$ | 75.8% | 78.40% ± 1.45% |
| $CL_{R2}$ | 81.91% | 82.40% ± 1.34% |

Our classifier used a mixture of the two training data sets described in Section 3. In Table 11, we compare the performance of our system on the following training data sets:

| | |
|---|---|
| *UNAMB*: | the data set of unambiguous examples described in Section 3.2 |
| *EM0*: | the data set of Section 3.1 after frequency table initialization |
| *EM1*: | *EM0* + one iteration of algorithm 3.1 |
| *EM2*: | *EM0* + two iterations of algorithm 3.1 |
| *EM3*: | *EM0* + three iterations of algorithm 3.1 |
| *1/8-EM1*: | one eighth of the data in *EM1* |
| *MIX*: | The concatenation of *UNAMB* and *EM1* |

Table 11 illustrates a slight but consistent increase in performance when using contextually similar words. However, since the confidence intervals overlap, we cannot claim with certainty

*Table 11*. Performance comparison of different data sets.

| DATABASE | ACCURACY WITHOUT SIMWORDS (95% CONF) | ACCURACY WITH SIMWORDS (95% CONF) |
|---|---|---|
| UNAMBIGUOUS | 83.15% ± 1.32% | 83.60% ± 1.30% |
| EM0 | 82.24% ± 1.35% | 82.69% ± 1.33% |
| EM1 | 83.76% ± 1.30% | 83.92% ± 1.29% |
| EM2 | 83.66% ± 1.30% | 83.70% ± 1.31% |
| EM3 | 83.20% ± 1.32% | 83.20% ± 1.32% |
| 1/8-EM1 | 82.98% ± 1.32% | 83.15% ± 1.32% |
| MIX | 84.11% ± 1.29% | 84.31% ± 1.28% |

*Table 12*. Performance with removal of *the* as $N_1$ or $N_2$.

| DATA SET | ACCURACY WITHOUT SIMWORDS (95% CONF) | ACCURACY WITH SIMWORDS (95% CONF) |
|---|---|---|
| WITH *THE* | 84.11% ± 1.29% | 84.31% ± 1.32% |
| WITHOUT *THE* | 84.44% ± 1.31% | 84.65% ± 1.30% |

that the contextually similar words improve performance.

In Section 7.1, we mentioned some testing examples contained $N_1$ = *the* or $N_2$ = *the*. For supervised algorithms, *the* is represented in the training set as any other noun. Consequently, these algorithms collect training data for *the* and performance is not affected. However, unsupervised methods break down on such examples. In Table 12, we illustrate the performance increase of our system when removing these erroneous examples.

## Conclusion and Future Work

The algorithms presented in this paper advance the state of the art for unsupervised approaches to prepositional phrase attachment and draws near the performance of supervised methods.

Currently, we are exploring different functions for combining contextually similar word approximations with the attachment scores. A promising approach considers the mutual information between the prepositional relationship of candidate attachments and $N_2$. As the mutual information decreases, our confidence in the attachment score decreases and the contextually similar word approximation is weighted higher. Also, improving the construction algorithm for contextually similar words would possibly improve the accuracy of the system. One approach first clusters the similar words. Then, dependency relationships are used to select the most representative clusters as the contextually similar words. The assumption is that more representative similar words produce better approximations.

## Acknowledgements

The authors wish to thank the reviewers for their helpful comments. This research was partly supported by Natural Sciences and Engineering Research Council of Canada grant OGP121338 and scholarship PGSB207797.

## References

Altmann, G. and Steedman, M. 1988. Interaction with Context During Human Sentence Processing. *Cognition*, 30:191-238.

Brill, E. 1995. Transformation-based Error-driven Learning and Natural Language Processing: A case study in part of speech tagging. *Computational Linguistics*, December.

Brill, E. and Resnik. P. 1994. A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of COLING-94*. Kyoto, Japan.

Collins, M. and Brooks, J. 1995. Prepositional Phrase Attachment through a Backed-off Model. In *Proceedings of the Third Workshop on Very Large Corpora*, pp. 27-38. Cambridge, Massachusetts.

Hindle, D. and Rooth, M. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1):103-120.

Lin, D. 1999. Automatic Identification of Non-Compositional Phrases. In *Proceedings of ACL-99*, pp. 317-324. College Park, Maryland.

Lin, D. 1998a. Extracting Collocations from Text Corpora. *Workshop on Computational Terminology*. Montreal, Canada.

Lin, D. 1998b. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING-ACL98*. Montreal, Canada.

Lin, D. (1994). Principar - an Efficient, Broad-Coverage, Principle-Based Parser. In *Proceedings of COLING-94*. Kyoto, Japan.

Miller, G. 1990. Wordnet: an On-Line Lexical Database. *International Journal of Lexicography*, 1990.

Ratnaparkhi, A. 1998. Unsupervised Statistical Models for Prepositional Phrase Attachment. In *Proceedings of COLING-ACL98*. Montreal, Canada.

Ratnaparkhi, A., Reynar, J., and Roukos, S. 1994. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 250-255. Plainsboro, N.J.

Roth, D. 1998. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proceedings of AAAI-98*, pp. 806-813. Madison, Wisconsin.

Stetina, J. and Nagao, M. 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.