A Nearest-Neighbor Method for Resolving PP-Attachment Ambiguity

Shaojun Zhao and **Dekang Lin**
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8
{shaojun,lindek}@cs.ualberta.ca

Abstract

We present a nearest-neighbor algorithm for resolving prepositional phrase attachment ambiguities. Its performance is significantly higher than previous corpus-based methods for PP-attachment that do not rely on manually constructed knowledge bases. We will also show that the PP-attachment task provides a way to evaluate methods for computing distributional word similarities. Our experiments indicate that the cosine of pointwise mutual information vector is a significantly better similarity measure than several other commonly used similarity measures.

1 Introduction

Natural language sentences often contain various forms of attachment ambiguities. Consider the following examples:

(1) PP-attachment ambiguity
   a. *Mary* [*ate* [*the salad*] [*with a fork*]]
   b. *Mary* [*ate* [*the salad* [*with croutons*]]

(2) Relative clause attachment ambiguity
   a. *They denied* [*the petition* [*for his release*] [*that was signed by over 10,000 people*]]
   b. *They denied* [*the petition for* [*his release* [*that would trigger an outrage*]]

(3) Pre-nominal modifier ambiguity
   a. [[*Child abuse*] *expert*]
   b. [*Child* [*computer expert*]]

Such ambiguities are a major source of parser errors (Lin 2003). In languages with little or no overt case marking, such as English and Chinese, syntactic information is insufficient to make attachment decisions (as demonstrated in the above examples). Many researchers have investigated the resolution of attachment ambiguities, especially for PP attachments.

As in previous work on PP-attachment (Hindle and Rooth 1993), we simplified the PP-Attachment problem by defining it as the following binary 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 the 4-tuple as either adverbial attachment (attaching to V) or adjectival attachment (attaching to N_1). For example, the 4-tuple (*eat*, *salad*, *with*, *fork*) is extracted from example (1a) and it has the target classification V; the 4-tuple (*eat*, *salad*, *with*, *croutons*) is extracted from example (1b) and it has the target classification N.

We present a nearest-neighbor algorithm for resolving PP-attachment ambiguities. The training examples are a set of 4-tuples with target classifications. Given a new 4-tuple to be classified, we search the training examples for its top-k nearest neighbors and determine its attachment based on the known classifications of the nearest neighbors. The similarity between two 4-tuples is determined by the distributional similarity between the corresponding words in the 4-tuples.

In the next section, we will briefly review the related work in PP-attachment. Section 3 presents several algorithms for computing distributional word similarity that are used in our experiments. The nearest-neighbor algorithm for resolving PP-attachment ambiguities is presented in Section 4. We then describe our experimental set up and results in Section 5 and conclude in Section 6.

2 Related Work

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

ating 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. They experimented with both word features and word class features. Their combination yielded 81.6% accuracy.

Collins and Brooks (1995) proposed a probabilistic model for PP-attachment, which employed a backed-off model to smooth the probabilities of unseen events. They discovered that P is the most informative lexical item for attachment disambiguation and keeping low frequency events increases performance. Their algorithm achieved 84.5% accuracy on the same data set as (Ratnaparkhi et al. 1994) and (Brill and Resnik 1994). This level of performance was also obtained by several methods proposed later. The boosting algorithm in (Abney *et al*. 1999) had an accuracy of 84.6%. The nearest-neighbor algorithm in (Zavrel *et al.* 1997) was 84.4% accurate.

The method in (Zavrel *et al.* 1997) is quite similar to ours. They construct a reduced-dimension feature vector for each word by applying the Principle Component Analysis (PCA). Each 4-tuple is represented by the concatenation of the feature vectors of the words in the 4-tuple. The distance between two 4-tuples is defined as the sum of the distances of each vector component.

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 algorithm in (Stetina and Nagao, 1997) employed a semantically tagged corpus. Each word in a labeled 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). Li (2002) reported 88.2% accuracy. However, his result was obtained with a different corpus than (Ratnaparkhi *et al*., 1994) and therefore is not directly comparable with ours.

3 Distributional Word Similarity

Words that tend to appear in the same contexts tend to have similar meanings. This is known as the Distributional Hypothesis in linguistics (Harris 1968). For example, the words *test* and *exam* are similar because both of them can be objects of verbs such as *administer, cancel, cheat on, conduct*… and both of them can be modified by adjectives such as *academic, comprehensive, diagnostic, difficult*…

Many methods have been proposed to compute distributional similarity between words, e.g., (Hindle, 1990), (Pereira *et al*. 1993), (Grefenstette 1994) and (Lin 1998). Almost all of the methods represent a word by a feature vector where each feature corresponds to a type of context in which the word appeared. They differ in how the feature vectors are constructed and how the similarity between two feature vectors is computed. The remainder of this section details the feature representation schemes and similarity formulas used in our experiments.

3.1 Syntax-based Feature Representation

In syntax-based method, the corpus is first parsed and then features are defined in terms of the dependency relationships between words in a corpus. A dependency relationship (Hays, 1964; Hudson, 1984; Mel'čuk, 1987) is an asymmetric binary relationship between a word called **head**, and another word called **modifier**. The structure of a sentence can be represented by a set of dependency relationships that form a tree. A word in the sentence may have several modifiers, but each word may modify at most one word. The root of the dependency tree does not modify any word. It is also called the head of the sentence.

For example, the following diagram shows the dependency tree for the sentence "*John found a solution to the problem*".

John found a solution to the problem.

The links in the diagram represent dependency relationships. The direction of a link is from the head to the modifier in the relationship. Labels associated with the links represent types of dependency relations.

We define a feature of a word *w* to be (*r, w'*) such that there is a dependency relationship *r* between *w* and *w'*. For example, the word *solution* in the above example has three features (*-obj, find*), (*to, problem*) and (*det, a*), where *-obj* is the inverse relation of the *obj* relation. The feature vectors of words are constructed by extracting all the features of all the words from the corpus. The value of each feature is the frequency count of the feature in the corpus. The following is a subset of the features and their values for the word *seminar*:

Features	Values
(*-subj, teach*)	3
(*-obj, sponsor*)	44
(*-obj, attend*)	81
(*-obj, teach*)	12
……	……

The features mean that *seminar* was used as the subject of *teach* 3 times, as the object of *sponsor* 44 times and as the object of *teach* 12 times.

3.2 Proximity-based Feature Representation

The syntax-based method requires a parsed corpus, which is often unavailable. Since most dependency relationships involve words that are situated close to one another, the dependency relationships can often be approximated by co-occurrence relationships within a small window. In previous approaches, the size of the window varies from 1 to the size of a document. However, having window size larger than 1 often introduce confusion between word association and word similarity. For example, suppose the window is ±5 words and the corpus contains a sentence *He died in a horrific traffic accident,* the words *died, in, a*, and *horrific* all become common features of *traffic* and *accident*. As a result, *traffic* and *accident* have a high similarity simply because they tend to collocate with one another.

In (Turney 2001) and (Terra and Clarke 2003), only one word within a window around *w* is chosen as a feature of *w*. They chose the word that has the highest pointwise mutual information with *w*. We have taken a different approach. We define features of *w* to be the first non-stop word on either side of *w* and the intervening stop words (which we arbitrarily define as the top-20 most frequent words in the corpus). For example, suppose the corpus consist of a single sentence *John found a solution to the problem*, the features of *solution* will be

Features	Values
(*left, find*)	0.50
(*left, a*)	0.50
(*right, to*)	0.33
(*right, the*)	0.33
(*right, problem*)	0.33

3.3 Similarity Measures

Once the contexts of a word are represented as a feature vector, the similarity between two words can be computed using their context vectors. We use $(u_1, u_2, ..., u_n)$ and $(v_1, v_2, ..., v_n)$ to denote the feature vectors for the word *u* and *v* respectively, where *n* is the number of feature types extracted from a corpus. We use f_i to denote the *i*th feature. The remainder of this subsection explains the similarity measures used in our experiments.

Cosine Measure

The Cosine Similarity Measure between two words computes the cosine between their feature vectors.

$$sim_{Cos}(u,v) = \frac{\sum_{i=1}^{n} u_i \times v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \times \sqrt{\sum_{i=1}^{n} v_i^2}}$$

A serious problem with the cosine measure is that the contribution of a feature in the similarity function is weighted by its frequency count. Features involving frequent words, such as *the* or *of*, tend to be given much higher weight than words with more semantic content.

Cosine of Pointwise Mutual Information

The pointwise mutual information between a feature f_i and a word u measures the strength association between them. It is defined as follows:

$$pmi(f_i,u)=\log\left(\frac{P(f_i,u)}{P(f_i)\times P(u)}\right)$$

where $P(f_i,u)$ is the probability of f_i co-occurring with u; $P(f_i)$ is the probability of f_i co-occurring with any word; and $P(u)$ is the probability of any feature co-occurring with u. The Cosine of Pointwise Mutual Information is defined as:

$$sim_{CosPMI}(u,v)=\frac{\sum_{i=1}^{n} pmi(f_i,u)\times pmi(f_i,v)}{\sqrt{\sum_{i=1}^{n} pmi(f_i,u)^2}\times\sqrt{\sum_{i=1}^{n} pmi(f_i,v)^2}}$$

Dice Co-efficient

The Dice co-efficient is a particularly simple similarity measure. It only distinguishes between zero and non-zero frequency counts.

$$sim_{Dice}(u,v)=\frac{2\times\sum_{i=1}^{n} s(u_i)\times s(v_i)}{\sum_{i=1}^{n} s(u_i)+\sum_{i=1}^{n} s(v_i)}$$

where $s(x)$=1 if x>0 and $s(x)$=0 otherwise.

Jensen-Shannon Divergence

The feature vectors can be converted into probability distributions by dividing each component with the sum of all components in the vector. The Jensen-Shannon divergence measures the distance between two probability distributions and their average distribution (Rao, 1982; Lin, 1991):

$$JS(q,r)=\frac{1}{2}\left[D(q\parallel avg_{q,r})+D(r\parallel avg_{q,r})\right]$$

where the function $D(p_1(x)\parallel p_2(x))$ is the KL-Divergence:

$$D(p_1(x)\parallel p_2(x))=\sum_x p_1(x)\log\left(\frac{p_1(x)}{p_2(x)}\right)$$

and $avg_{q,r}$ is the average distribution of q and r:

$$avg_{q,r}(x)=\frac{1}{2}\left[q(x)+r(x)\right]$$

The Jensen-Shannon divergence can be converted into a similarity measure by defining

$$sim_{JS}(u,v)=\frac{1}{1+JS(u,v)}$$

When u and v are identical distributions, $JS(u,v)=0$ and $sim_{JS}(u,v)=1$.

4 Resolving PP-Attachment Ambiguity with Nearest Neighbors

Our algorithm for resolving PP-attachment ambiguity takes a 4-tuple (V, N_1, P, N_2) as input and classifies it as N or V. The class N means that the prepositional phrase modifies the noun N_1. The class V means that the prepositional phrase modifies the verb V. The classification process is comprised of 4 steps. Each step attempts to make a decision according to the weighted majority vote by the nearest neighbors of the input. The weight of a training example is the similarity between it and the input. Different steps have different definitions of what constitutes the nearest neighbors and use different formulas to compute the similarity between two 4-tuples. A step is taken only if the previous step fails to make a decision, which happens when the classes N and V received an equal vote (often due to an empty set of nearest neighbors).

In **Step 1**, the nearest neighbors consist of training examples that are identical to the input 4-tuple. The similarity between such an example and the input is 1.

In **Step 2**, the nearest neighbors consist of the top-k (k is determined using a development data set) most similar training examples that involve the same preposition as the input. The similarity between two 4-tuples t_1 and t_2 is computed as follows:

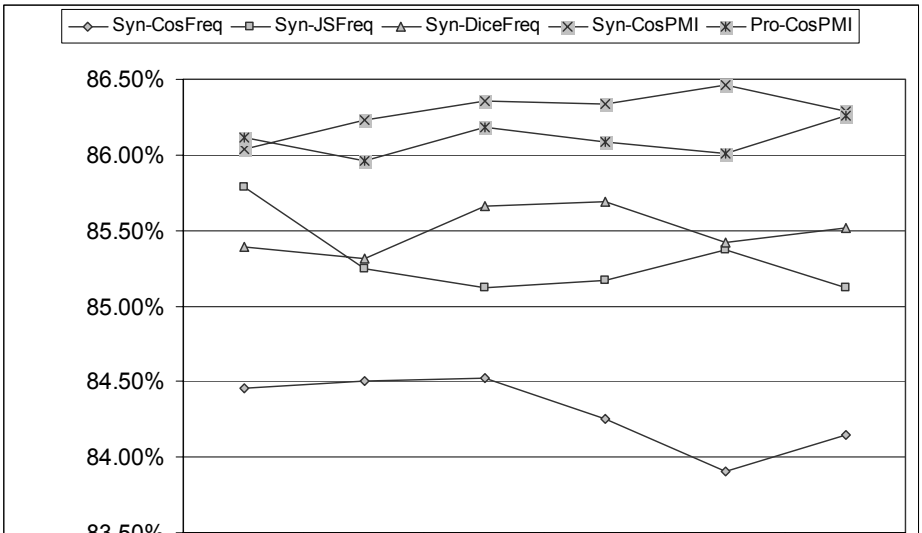
$$\text{sim}(t_1, t_2)=ab+bc+ca$$

where a, b and c are the distributional word similarities between the two verbs, the two N_1's and the two N_2's, respectively. The word similarity can be computed using any of the similarity measures discussed in the previous section. The 4-tuple similarity sim(t_1, t_2) will be zero when any two of a, b and c are zeros.

Step 3 is identical to Step 2, except that the similarity between two 4-tuples t_1 and t_2 is defined as sim(t_1, t_2)=a+b+c, which has a zero value only when all of a, b and c are zeros.

In **Step 4**, the set of nearest neighbors includes all the training examples that involve the same preposition P as the input. The similarity between each of them and the input is considered to be a constant. If the votes for N and V are tied in this step, the default classification N is assigned to the input.

Figure 1: Parameter Tuning on Development Set



	11	15	25	29	33	35
Syn-CosFreq	84.45%	84.50%	84.53%	84.25%	83.91%	84.15%
Syn-JSFreq	85.79%	85.24%	85.12%	85.17%	85.37%	85.12%
Syn-DiceFreq	85.39%	85.32%	85.66%	85.69%	85.42%	85.52%
Syn-CosPMI	86.04%	86.23%	86.36%	86.33%	86.46%	86.28%
Pro-CosPMI	86.11%	85.96%	86.18%	86.09%	86.01%	86.26%

5 Experimental Results and Discussions

Data Sets

We computed the word similarities using the ACQUAINT Corpus, which consists of newswire text data in English, drawn from three sources: the Xinhua News Service (People's Republic of China), the New York Times News Service, and the Associated Press Worldstream News Service. It contains roughly 375 million words (about 3 GB of text). We computed the distributional word similarity using both syntax and proximity based methods. For the syntax-based methods, we parsed the corpus with Minipar (Lin 1994) and extracted dependency relationships. An evaluation with the Susanne corpus showed that 88.54% of the dependencies identified by Minipar are correct (Lin 2003).

The PP-attachment data set used in our experiments was extracted from the Penn Treebank WSJ corpus by Ratnaparkhi *et al* (1994). It includes a training set, a development set and a testing set, consisting of 20801, 4039 and 3097 elements respectively. We preprocessed the data to reduce all the words to their root forms. We replaced digits with @ signs. For example, 2003 becomes @@@@; $1.15 becomes $@.@@.

Parameter Tuning

For each feature type and similarity measure, we run the nearest neighbor algorithm on the development data set with different k. Figure 1 shows the results. We chose the k-values that produced the best performance on the development set and used them in the experiments with the test set.

Results on the Test Data Set

Table 1 shows the result of PP attachment accuracy using different methods for computing word similarity. Among different similarity measures, sim_{CosPMI} has the best performance, which is significantly higher than other similarity measures listed in this table with ≥95% confidence (assuming binomial distribution). It can also be seen that syntax-based features are better than proximity-based ones.

Despite the high performance of sim_{CosPMI}, the closely related sim_{Cos} measure performed rather poorly. The only difference between the two is that sim_{Cos} uses the frequency counts as feature values

whereas sim_{CosPMI} uses the pointwise mutual information between a feature and the word. This suggests that frequency counts are inappropriate as feature values. In fact, the sim_{Dice} measure, which totally ignores all frequency counts, outperforms the measures that use them: sim_{Cos} and $sim_{JS.}$

Table 1: PP-Attachment Accuracy

Feature Type	Similarity	Accuracy	k
Syntax-based	sim_{Cos}	83.1%	25
Syntax-based	sim_{JS}	84.9%	11
Syntax-based	sim_{Dice}	85.3%	29
Syntax-based	sim_{CosPMI}	**86.5%**	33
Proximity-based	sim_{CosPMI}	85.6%	35

The accuracies achieved by previous corpus-based methods that do not rely on manually constructed knowledge bases are shown in Table 2. The 86.5% accuracy by sim_{CosPMI} with syntax-based features is significantly higher than all of them with ≥98% confidence (assuming errors are binomially distributed).

Table 2: Previous PP-Attachment Accuracies

Method	Accuracy
Ratnaparkhi, Reynar and Roukos 94	81.6%
Collins and Brooks 95	84.5%
Zavrel, Daelemans and Veenstra 97	84.4%
Abney, Schapire and Singer 99	84.6%
Our Result (using sim_{CosPMI})	86.5%

For each of the steps in our algorithm, Table 3 gives its coverage (the number of decisions made in that step), the coverage % (the percentage of decisions made in that step) and the accuracy (the percentage of correct decisions in that step). It can be seen that over 99.8% of the coverage is provided by Steps 1 and 2 in which the preposition *P* and at least two of *V*, N_1 and N_2 are involved in the decisions. In contrast, only 40% of the decisions involve as many words in the method proposed in (Collins and Brooks 1994).

Table 3: Coverage and Accuracy of 4 Steps

Step	Coverage	Coverage %	Accuracy
1	244	7.88%	91.8%
2	2848	91.96%	86.0%
3	2	0.06%	100.0%
4	3	0.10%	100.0%
Overall	3097	100.00%	86.5%

6 Conclusion

We presented a nearest-neighbor algorithm for resolving prepositional phrase attachment ambiguities. Our algorithm achieved a significantly higher accuracy than other corpus-based methods that do not rely on manually constructed knowledge bases. We also demonstrated that the PP-attachment problem can also be used as an evaluation criterion for distributional word similarity. Our experiments showed that the cosine of pointwise mutual information is significantly better than several other commonly used similarity measures.

7 Acknowledgements

The authors wish to thank the anonymous reviewers for valuable inputs. This research is funded by NSERC Research Grant OGP121338 and by The Alberta Ingenuity Centre for Machine Learning.

References

Abney, S and Schapire, R.E. and Singer, Y. 1999. Boosting Applied to Tagging and PP-attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP-VLC*, pages 38--45, College Park, MD.

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*. pp.1198-1204. 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.

Grefenstette, G. 1994. *Explorations in Automatic Thesaurus Discovery*, Kluwer Academic Publishers, Boston.

Harris, Z.S.1968. *Mathematical Structures of Language*. New York: Wiley.

Hays, D. 1964. Dependency Theory: a Formalism and Some Observations. *Language*, 40:511-525.

Hindle, D. 1990. Noun Classification from Predicate-Argument Structures. In *Proceedings of ACL-90*. pp. 268-275. Pittsburgh, Pennsylvania.

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

Hudson, R. 1984. *Word Grammar*. Basil Blackwell Publishers Limited. Oxford, England.

Li, H. 2002. Word Clustering and Disambiguation based on Co-occurrence Data. *Natural Language Engineering*, 8(1), 25-42.

Lin, D. 1998. 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.

Lin, D. 2003. Dependency-based evaluation of MINIPAR. In *Building and using syntactically annotated corpora*, Anne Abeille (editor), pp. 317-330. KLUWER, Dordrecht.

Lin, J. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145{151.

Mel'čuk, I. A. 1987. *Dependency Syntax: theory and practice*. State University of New York Press. Albany, NY.

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

Pereira, F., Tishby, N., and Lee, L. 1993. Distributional Clustering of English Words. In *Proceedings of ACL-93*. pp. 183-190. Columbus, Ohio.

Rao, C.R. 1982. Diversity: Its measurement, decomposition, apportionment and analysis. *Sankyha: The Indian Journal of Statistics*, 44(A):1{22.

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.

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.

Terra, E. L. and Clarke, C. 2003. Frequency Estimates for Statistical Word Similarity Measures. In *the Proceedings of the 2003 Human Language Technology Conference*, pp.244-251. Edmonton, Canada, May.

Turney, P.D. (2001), Mining the Web for synonyms: PMI-IR versus LSA on TOEFL, *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, Freiburg, Germany, pp. 491-502.

Zavrel, J. and Daelemans, W and Veenstra, J. 1997. Resolving PP attachment Ambiguities with Memory-Based Learning. In *Proceedings of the Conference on Computational Natural Language Learning, CoNLL97*, pages 136-144, Madrid, Spain.