PP-Attachment Disambiguation Boosted by a Gigantic Volume of Unambiguous Examples

Daisuke Kawahara and Sadao Kurohashi

Graduate School of Information Science and Technology, University of Tokyo,
7-3-1 Hongo Bunkyo-ku, Tokyo, 113-8656, Japan
{kawahara, kuro}@kc.t.u-tokyo.ac.jp

Abstract. We present a PP-attachment disambiguation method based on a gigantic volume of unambiguous examples extracted from raw corpus. The unambiguous examples are utilized to acquire precise lexical preferences for PP-attachment disambiguation. Attachment decisions are made by a machine learning method that optimizes the use of the lexical preferences. Our experiments indicate that the precise lexical preferences work effectively.

1 Introduction

For natural language processing (NLP), resolving various ambiguities is a fundamental and important issue. Prepositional phrase (PP) attachment ambiguity is one of the structural ambiguities. Consider, for example, the following sentences [1]:

(1) a. Mary ate the salad with a fork.

b. Mary ate the salad with croutons.

The prepositional phrase in (1a) "with a fork" modifies the verb "ate", because "with a fork" describes how the salad is eaten. The prepositional phrase in (1b) "with croutons" modifies the noun "the salad", because "with croutons" describes the salad. To disambiguate such PP-attachment ambiguity, some kind of world knowledge is required. However, it is currently difficult to give such world knowledge to computers, and this situation makes PP-attachment disambiguation difficult. Recent state-of-the-art parsers perform with the practical accuracy, but seem to suffer from the PP-attachment ambiguity [2, 3].

For NLP tasks including PP-attachment disambiguation, corpus-based approaches have been the dominant paradigm in recent years. They can be divided into two classes: supervised and unsupervised. Supervised methods automatically learn rules from tagged data, and achieve good performance for many NLP tasks, especially when lexical information, such as words, is given. Such methods, however, cannot avoid the sparse data problem. This is because tagged data are not sufficient enough to discriminate a large variety of lexical information. To deal with this problem, many smoothing techniques have been proposed.

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 188–198, 2005.

The other class for corpus-based approaches is unsupervised learning. Unsupervised methods take advantage of a large number of data that are extracted from large raw corpora, and thus can alleviate the sparse data problem. However, the problem is their low performance compared with supervised methods, because of the use of unreliable information.

For PP-attachment disambiguation, both supervised and unsupervised methods have been proposed, and supervised methods have achieved better performance (e.g., 86.5% accuracy by [1]). Previous unsupervised methods tried to extract reliable information from large raw corpora, but the extraction heuristics seem to be inaccurate [4, 5]. For example, Ratnaparkhi extracted unambiguous word triples of (verb, preposition, noun) or (noun, preposition, noun), and reported that their accuracy was 69% [4]. This means that the extracted triples are not truly unambiguous, and this inaccurate treatment may have led to low PP-attachment performance (81.9%).

This paper proposes a PP-attachment disambiguation method based on an enormous amount of truly unambiguous examples. The unambiguous examples are extracted from raw corpus using some heuristics inspired by the following example sentences in [6]:

(2) a. She sent him into the nursery to gather up his toys.
b. The road to London is long and winding.

In these sentences, the underlined PPs are unambiguously attached to the double-underlined verb or noun. The extracted unambiguous examples are utilized to acquire precise lexical preferences for PP-attachment disambiguation. Attachment decisions are made by a machine learning technique that optimizes the use of the lexical preferences. The point of our work is to use a "gigantic" volume of "truly" unambiguous examples. The use of only truly unambiguous examples leads to statistics of high-quality and good performance of disambiguation in spite of the learning from raw corpus. Furthermore, by using a gigantic volume of data, we can alleviate the influence of the sparse data problem.

The remainder of this paper is organized as follows. Section 2 briefly describes the globally used training and test set of PP-attachment. Section 3 summarizes previous work for PP-attachment. Section 4 describes a method of calculating lexical preference statistics from a gigantic volume of unambiguous examples. Section 5 is devoted to our PP-attachment disambiguation algorithm. Section 6 presents the experiments of our disambiguation method. Section 7 gives the conclusions.

2 Tagged Data for PP-Attachment

The PP-attachment data with correct attachment site are available [1]. These data were extracted from Penn Treebank [7] by the IBM research group [8]. Hereafter, we call these data "IBM data". Some examples in the IBM data are shown in Table 1.

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

Table 1. Some Examples of the IBM data

v	n_1	p	n_2	attach
join	board	as	director	V
is	chairman	of	N.V.	N
using	crocidolite	in	filters	V
bring	attention	to	problem	V
is	asbestos	in	product	N
making	paper	for	filters	N
including	three	with	cancer	N

Table 2. Various Baselines and Upper Bounds of PP-Attachment Disambiguation

method	accuracy
always N	59.0%
N if p is "of"; otherwise V	70.4%
most likely for each preposition	72.2%
average human (only quadruple)	88.2%
average human (whole sentence)	93.2%

The data consist of 20,801 training and 3,097 test tuples. In addition, a development set of 4,039 tuples is provided. Various baselines and upper bounds of PP-Attachment disambiguation are shown in Table 2. All the accuracies except the human performances are on the IBM data. The human performances were reported by [8].

3 Related Work

There have been lots of supervised approaches for PP-attachment disambiguation. Most of them used the IBM data for their training and test data.

Ratnaphakhi et al. proposed a maximum entropy model considering words and semantic classes of quadruples, and performed with 81.6% accuracy [8]. Brill and Resnik presented a transformation-based learning method [9]. They reported 81.8% accuracy, but they did not use the IBM data [2]. Collins and Brooks used a probabilistic model with backing-off to smooth the probabilities of unseen events, and its accuracy was 84.5% [10]. Stetina and Nagao used decision trees combined with a semantic dictionary [11]. They achieved 88.1% accuracy, which is approaching the human accuracy of 88.2%. This great performance is presumably indebted to the manually constructed semantic dictionary, which can be regarded as a part of world knowledge. Zavrel et al. employed a nearest-neighbor method, and its accuracy was 84.4% [12]. Abney et al. proposed a boosting approach, and yielded 84.6% accuracy [13]. Vanschoenwinkel and Manderick introduced a kernel method into PP-attachment disam-

[2] The accuracy on the IBM data was 81.9% [10].

biguation, and attained 84.8% accuracy [14]. Zhao and Lin proposed a nearest-neighbor method with contextually similar words learned from large raw corpus [1]. They achieved 86.5% accuracy, which is the best performance among previous methods for PP-attachment disambiguation without manually constructed knowledge bases.

There have been several unsupervised methods for PP-attachment disambiguation. Hindle and Rooth extracted over 200K (v, n_1, p) triples with ambiguous attachment sites from 13M words of AP news stories [15]. Their disambiguation method used lexical association score, and performed at 75.8% accuracy on their own data set. Ratnaparkhi collected 910K unique unambiguous triples (v, p, n_2) or (n_1, p, n_2) from 970K sentences of Wall Street Journal, and proposed a probabilistic model based on cooccurrence values calculated from the collected data [4]. He reported 81.9% accuracy. As previously mentioned, the accuracy was possibly lowered by the inaccurate (69% accuracy) extracted examples. Pantel and Lin extracted ambiguous 8,900K quadruples and unambiguous 4,400K triples from 125M word newspaper corpus [5]. They utilized scores based on cooccurrence values, and resulted in 84.3% accuracy. The accuracy of the extracted unambiguous triples are unknown, but depends on the accuracy of their parser.

There is a combined method of supervised and unsupervised approaches. Volk combined supervised and unsupervised methods for PP-attachment disambiguation for German [16]. He extracted triples that are possibly unambiguous from 5.5M words of a science magazine corpus, but these triples were not truly unambiguous. His unsupervised method is based on cooccurrence probabilities learned from the extracted triples. His supervised method adopted the backed-off model by Collins and Brooks. This model is learned the model from 5,803 quadruples. Its accuracy on a test set of 4,469 quadruples was 73.98%, and was boosted to 80.98% by the unsupervised cooccurrence scores. However, his work was constrained by the availability of only a small tagged corpus, and thus it is unknown whether such an improvement can be achieved if a larger size of a tagged set like the IBM data is available.

4 Acquiring Precise Lexical Preferences from Raw Corpus

We acquire lexical preferences that are useful for PP-attachment disambiguation from a raw corpus. As such lexical preferences, cooccurrence statistics between the verb and the prepositional phrase or the noun and the prepositional phrase are used. These cooccurrence statistics can be obtained from a large raw corpus, but the simple use of such a raw corpus possibly produces unreliable statistics. We extract only truly unambiguous examples from a huge raw corpus to acquire precise preference statistics.

This section first mentions the raw corpus, and then describes how to extract truly unambiguous examples. Finally, we explain our calculation method of the lexical preferences.

4.1 Raw Corpus

In our approach, a large volume of raw corpus is required. We extracted raw corpus from 200M Web pages that had been collected by a Web crawler for a month [17]. To obtain the raw corpus, each Web page is processed by the following tools:

1. sentence extracting
   Sentences are extracted from each Web page by a simple HTML parser.
2. tokenizing
   Sentences are tokenized by a simple tokenizer.
3. part-of-speech tagging
   Tokenized sentences are given part-of-speech tags by Brill tagger [18].
4. chunking
   Tagged sentences are chunked by YamCha chunker [19].

By the above procedure, we acquired 1,300M chunked sentences, which consist of 21G words, from the 200M Web pages.

4.2 Extraction of Unambiguous Examples

Unambiguous examples are extracted from the chunked sentences. Our heuristics to extract truly unambiguous examples were decided in the light of the following two types of unambiguous examples in [6].

(3) a. She sent him into the nursery to gather up his toys.
    b. The road to London is long and winding.

The prepositional phrase "into the nursery" in (3a) must attach to the verb "sent", because attachment to a pronoun like "him" is not possible. The prepositional phrase "to London" in (3b) must attach to the noun "road", because there are no preceding possible heads.

We use the following two heuristics to extract unambiguous examples like the above.

- To extract an unambiguous triple (v, p, n_2) like (3a), a verb followed by a pronoun and a prepositional phrase is extracted.
- To extract an unambiguous triple (n_1, p, n_2) like (3b), a noun phrase followed by a prepositional phrase at the beginning of a sentence is extracted.

4.3 Post-processing of Extracted Examples

The extracted examples are processed in the following way:

- For verbs (v):
  - Verbs are reduced to their lemma.
- For nouns (n_1, n_2):
  - 4-digit numbers are replaced with <year>.

- All other strings of numbers were replaced with <num>.
- All words at the beginning of a sentence are converted into lower case.
- All words starting with a capital letter followed by one or more lower case letters were replaced with <name>.
- All other words are reduced to their singular form.

– For prepositions (p):
- Prepositions are converted into lower case.

As a result, 21M (v, p, n_2) triples and 147M (n, p, n_2) triples,in total 168M triples, were acquired.

4.4 Calculation of Lexical Preferences for PP-Attachment

From the extracted truly unambiguous examples, lexical preferences for PP-attachment are calculated. As the lexical preferences, pointwise mutual information between v and "p n_2" is calculated from cooccurrence counts of v and "p n_2" as follows[3]:

$$I(v, pn_2) = \log \frac{\frac{f(v,pn_2)}{N}}{\frac{f(v)}{N}\frac{f(pn_2)}{N}} \tag{1}$$

where N denotes the total number of the extracted examples (168M), $f(v)$ and $f(pn_2)$ is the frequency of v and "p n_2", respectively, and $f(v, pn_2)$ is the cooccurrence frequency of v and pn_2.

Similarly, pointwise mutual information between n_1 and "p n_2" is calculated as follows:

$$I(n_1, pn_2) = \log \frac{\frac{f(n_1,pn_2)}{N}}{\frac{f(n_1)}{N}\frac{f(pn_2)}{N}} \tag{2}$$

The preference scores ignoring n_2 are also calculated:

$$I(v, p) = \log \frac{\frac{f(v,p)}{N}}{\frac{f(v)}{N}\frac{f(p)}{N}} \tag{3}$$

$$I(n_1, p) = \log \frac{\frac{f(n_1,p)}{N}}{\frac{f(n_1)}{N}\frac{f(p)}{N}} \tag{4}$$

5 PP-Attachment Disambiguation Method

Our method for resolving PP-attachment ambiguity takes a quadruple (v, n_1, p, n_2) as input, and classifies it as V or N. The class V means that the prepositional

[3] As in previous work, simple probability ratios can be used, but a preliminary experiment on the development set shows their accuracy is worse than the mutual information by approximately 1%.

phrase "p n_2" modifies the verb v. The class N means that the prepositional phrase modifies the noun n_1.

To solve this binary classification task, we employ Support Vector Machines (SVMs), which have been well-known for their good generalization performance [20].

We consider the following features:

- LEX: word of each quadruple
  To reduce sparse data problems, all verbs and nouns are pre-processed using the method stated in Section 4.3.
- POS: part-of-speech information of v, n_1 and n_2
  POSs of v, n_1 and n_2 provide richer information than just verb or noun, such as inflectional information.
  The IBM data, which we use for our experiments, do not contain POS information. To obtain POS tags of a quadruple, we extracted the original sentence of each quadruple from Penn Treebank, and applied the Brill tagger to it. Instead of using the correct POS information in Penn Treebank, we use the POS information automatically generated by the Brill tagger to keep the experimental environment realistic.
- LP: lexical preferences
  Given a quadruple (v, n_1, p, n_2), four statistics calculated in Section4.4, $I(v, pn_2)$, $I(n_1, pn_2)$, $I(v, p)$ and $I(n_1, p)$, are given to SVMs as features.

6 Experiments and Discussions

We conducted experiments on the IBM data. As an SVM implementation, we employed SVM^{light} [21]. To determine parameters of SVM^{light}, we run our method on the development data set of the IBM data. As the result, parameter j, which is used to make much account of training errors on either class [22], is set to 0.65, and 3-degree polynomial kernel is chosen. Table 3 shows the experimental results for PP-attachment disambiguation. For comparison, we conducted several experiments with different feature combinations in addition to our proposed method "LEX+POS+LP", which uses all of the three types of features. The proposed method "LEX+POS+LP" surpassed "LEX", which is the standard supervised model, and furthermore, significantly outperformed all other

Table 3. PP-Attachment Accuracies

LEX	POS	LP	accuracy
√			85.34
√	√		85.05
		√	83.73
	√	√	84.66
√		√	86.44
√	√	√	**87.25**

Table 4. Precision and Recall for Each Attachment Site ("LEX+POS+LP" model)

class	precision	recall
V	1067/1258 (84.82%)	1067/1271 (83.95%)
N	1635/1839 (88.91%)	1635/1826 (89.54%)

Table 5. PP-Attachment Accuracies of Previous Work

	method	accuracy
our method	SVM	87.25%
supervised		
Ratnaphakhi et al., 1994	ME	81.6%
Brill and Resnik, 1994	TBL	81.9%
Collins and Brooks, 1995	back-off	84.5%
Zavrel et al., 1997	NN	84.4%
Stetina and Nagao, 1997	DT	88.1%
Abney et al., 1999	boosting	84.6%
Vanschoenwinkel and Manderick, 2003	SVM	84.8%
Zhao and Lin, 2004	NN	86.5%
unsupervised		
Ratnaparkhi, 1998	-	81.9%
Pantel and Lin, 2000	-	84.3%

ME: Maximum Entropy, TBL: Transformation-Based Learning,
DT: Decision Tree, NN: Nearest Neighbor

configurations (McNemar's test; $p < 0.05$). "LEX+POS" model was a little worse than "LEX", but "LEX+POS+LP" was better than "LEX+LP" (and also "POS+LP" was better than "LP"). From these results, we can see that "LP" worked effectively, and the combination of "LEX+POS+LP" was very effective. Table 4 shows the precision and recall of "LEX+POS+LP" model for each class (N and V).

Table 5 shows the accuracies achieved by previous methods. Our performance is higher than any other previous methods except [11]. The method of Stetina and Nagao employed a manually constructed sense dictionary, and this conduces to good performance.

Figure 1 shows the learning curve of "LEX" and "LEX+POS+LP" models while changing the number of tagged data. When using all the training data, "LEX+POS+LP" was better than "LEX"by approximately 2%. Under the condition of small data set, "LEX+POS+LP" was better than "LEX"by approximately 5%. In this situation, in particular, the lexical preferences worked more effectively.

Figure 2 shows the learning curve of "LEX+POS+LP" model while changing the number of used unambiguous examples. The accuracy rises rapidly by 10M unambiguous examples, and then drops once, but after that rises slightly. The best score 87.28% was achieved when using 77M unambiguous examples.

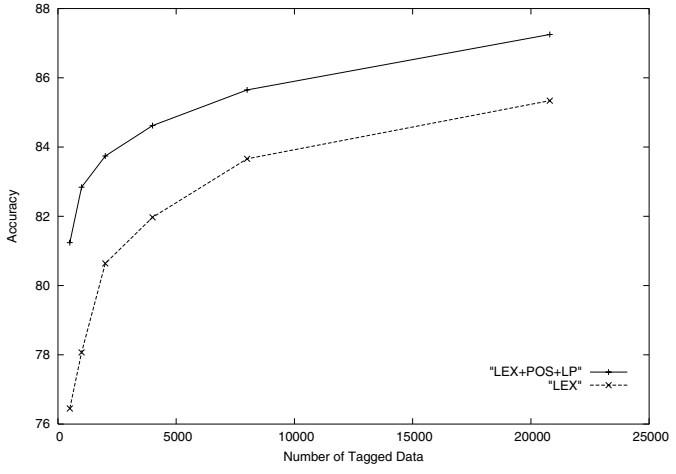

Fig. 1. Learning Curve of PP-Attachment Disambiguation

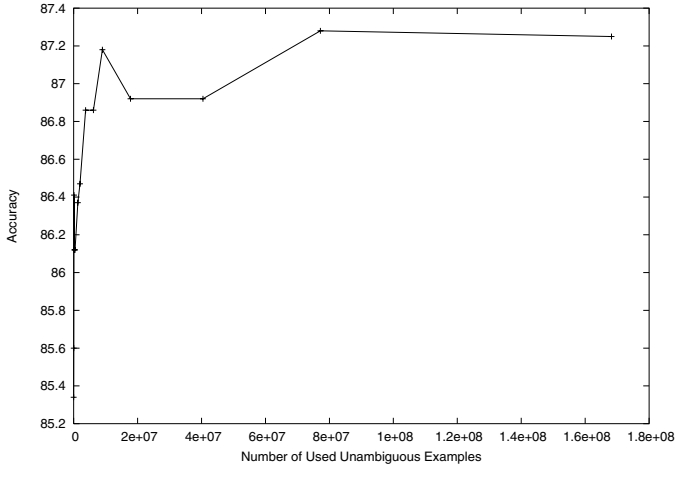

Fig. 2. Learning Curve of PP-Attachment Disambiguation while changing the number of used unambiguous examples

7 Conclusions

This paper has presented a corpus-based method for PP-attachment disambiguation. Our approach utilizes precise lexical preferences learned from a gigantic volume of truly unambiguous examples in raw corpus. Attachment decisions are made using a machine learning method that incorporates these lexical preferences. Our experiments indicated that the precise lexical preferences worked effectively.

In the future, we will investigate useful contextual features for PP-attachment, because human accuracy improves by around 5% when they see more than just a quadruple.

Acknowledgements

We would like to thank Prof. Kenjiro Taura for allowing us to use an enormous volume of Web corpus. We also would like to express our thanks to Tomohide Shibata for his constructive and fruitful discussions.

References

1. Zhao, S., Lin, D.: A nearest-neighbor method for resolving pp-attachment ambiguity. In: Proceedings of the 1st International Joint Conference on Natural Language Processing. (2004) 428–434
2. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania (1999)
3. Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics. (2000) 132–139
4. Ratnaparkhi, A.: Statistical models for unsupervised prepositional phrase attachment. In: Proceedings of the 17th International Conference on Computational Linguistics. (1998) 1079–1085
5. Pantel, P., Lin, D.: An unsupervised approach to prepositional phrase attachment using contextually similar words. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. (2000) 101–108
6. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
7. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics **19** (1994) 313–330
8. Ratnaparkhi, A., Reynar, J., Roukos, S.: A maximum entropy model for prepositional phrase attachment. In: Proceedings of the ARPA Human Language Technology Workshop. (1994) 250–255
9. Brill, E., Resnik, P.: A rule-based approach to prepositional phrase attachment disambiguation. In: Proceedings of the 15th International Conference on Computational Linguistics. (1994) 1198–1204
10. Collins, M., Brooks, J.: Prepositional phrase attachment through a backed-off model. In: Proceedings of the 3rd Workhop on Very Large Corpora. (1995) 27–38
11. Stetina, J., Nagao, M.: Corpus based pp attachment ambiguity resolution with a semantic dictionary. In: Proceedings of the 5th Workhop on Very Large Corpora. (1997) 66–80
12. Zavrel, J., Daelemans, W., Veenstra, J.: Resolving pp attachment ambiguities with memory-based learning. In: Proceedings of the Workshop on Computational Natural Language Learning. (1997) 136–144
13. Abney, S., Schapire, R., Singer, Y.: Boosting applied to tagging and pp attachment. In: Proceedings of 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. (1999) 38–45

14. Vanschoenwinkel, B., Manderick, B.: A weighted polynomial information gain kernel for resolving pp attachment ambiguities with support vector machines. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence. (2003) 133–138
15. Hindle, D., Rooth, M.: Structural ambiguity and lexical relations. Computational Linguistics **19** (1993) 103–120
16. Volk, M.: Combining unsupervised and supervised methods for pp attachment disambiguation. In: Proceedings of the 19th International Conference on Computational Linguistics. (2002) 1065–1071
17. Takahashi, T., Soonsang, H., Taura, K., Yonezawa, A.: World wide web crawler. In: Poster Proceedings of the 11th International World Wide Web Conference. (2002)
18. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics **21** (1995) 543–565
19. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics. (2001) 192–199
20. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
21. Joachims, T.: 11. In: Making Large-Scale Support Vector Machine Learning Practical, in Advances in Kernel Methods - Support Vector Learning. MIT Press (1999) 169–184
22. Morik, K., Brockhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In: Proceedings of the 16th International Conference on Machine Learning. (1999) 268–277