PREPOSITIONAL PHRASE ATTACHMENT AMBIGUITY RESOLUTION USING SEMANTIC HIERARCHIES

Kailash Nadh
School of Computing Science
Middlesex University
London, UK
email: k.nadh@mdx.ac.uk

Christian R. Huyck
School of Computing Science
Middlesex University
London, UK
email: c.huyck@mdx.ac.uk

Abstract

This paper describes a system that resolves prepositional phrase attachment ambiguity in English sentence processing. This attachment problem is ubiquitous in English text, and is widely known as a place where semantics determines syntactic form. The decision is made based on a four-tuple composed of the head verb of the verb phrase, the head noun of the noun phrase, and the preposition and head noun in the prepositional phrase. A corpus with known results, the Penn Treebank, is used for training and testing purposes. During training, known results are used to build a lattice of hierarchical categories taken from WordNet. These lattices are then compared to the novel lattices derived from the test four-tuples. The results of the system are 90.53% correct attachment decisions.

Keywords: Prepositional Phrase Attachment, Syntactic and Semantic ambiguity, Semantic Nets, Hierarchies, Parsing, Ambiguity Resolution

1 Introduction

Natural language parsing is a complex task that takes the words of a sentence as input and creates a representation of the sentence that typically includes the syntactic relations and frequently the semantic relations. Frequently sentences can have multiple syntactic and semantic interpretations, but when humans parse they rarely derive more than one interpretation. One common form of ambiguity is prepositional phrase (PP) attachment ambiguity.

One commonly used example of PP attachment ambiguity is (ex. 1):

(ex. 1) I saw the girl with the telescope.

In this particular sentence, people typically attach the PP directly to the verb phrase (VP) so that *the telescope* is the instrument. The other interpretation attaches the PP to the noun phrase (NP) so that *the telescope* modifies *the girl* with the corresponding semantics being something like *the girl* possesses *the telescope*.

PP attachment ambiguity is a problem for parsing. As it is syntactically ambiguous, to resolve the ambiguity properly, some form of semantics is needed. While even the entire semantics of a sentence are insufficient to solve the problem in every case, most of the ambiguities can be resolved by the four main components (1) head verb, (2) head noun of NP, (3) preposition, and (4) head noun of PP. Several studies have been conducted based on these four features (e.g. [12, 10]).

A relatively simple way to solve the problem is to use examples of PP attachment ambiguity where the result is known. Whenever these same four-tuples are seen again choose the same attachment; this will not always work, but will work most of the time. Unfortunately, the problem is sparseness of data; most four-tuples have never been seen before.

This paper describes a novel method based on the combination of the word sense hierarchies of the three open category items (verb, noun, noun) and the bare preposition. These are combined into a lattice [16]. Using a simple supervised training algorithm, examples extracted from the Penn Treebank [8] are used to populate the lattice. When trying to resolve a novel attachment, the lattice is searched from the item represented by the four-tuple. The results of this on a test on the Penn Treebank is 90.53%; to the best of the authors' knowledge, this is the best result of any computational system for PP attachment ambiguity resolution.

2 Background

Perhaps the main difficulty with natural language parsing is that the syntactic structure of a sentence is frequently ambiguous, and semantics is needed to resolve the ambiguity. There are a wide range of ambiguities including conjunctive ambiguity and ambiguity between relative clauses and main verbs, but PP attachment ambiguity is probably the most widely researched example of this need to resolve syntactic ambiguity. Moreover, as text mining and other applications of natural language processing become more pervasive, this problem moves from one of psychological and linguistic interest, to one of engineering and business.

2.1 PP Attachment Ambiguity

PP attachment ambiguity is widely discussed in the literature. In this section, a simple explanation is given. The explanation is consistent with most current syntactic theo-

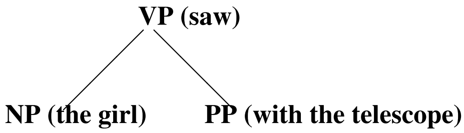

Figure 1. PP attached to the VP

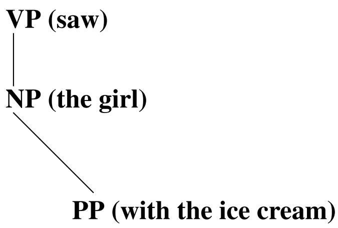

Figure 2. PP attached to the NP

ries [6], and is based on syntax trees. In English, the basic ambiguity occurs when the sequence *VP NP PP* occurs.

Figure 1 refers to the case where the PP is attached to the VP. The attached text refers to the standard interpretation of (ex. 1) with *the telescope* used as the instrument for seeing. Note that the actor of the sentence *NP I* is omitted from the diagram.

A second example is (ex. 2):

(ex. 2) I saw the girl with the ice cream.

Humans typically interpret this sentence with the PP attaching to the NP. The semantics is something like *the girl* has *the ice cream*. Figure 2 shows a syntax tree when the PP attaches to the noun with the text referring to (ex. 2).

2.2 Psycholinguistics of PP Attachment

It is clear that PP attachments can not be resolved solely from information contained in the sentence, because context influences the attachment decision. In the canonical example, people make the reverse attachment decision when the prior sentence is *There are two girls, one with a telescope and one with a hat*.

None the less, most studies are done on the null context, with only single sentences being presented. Even in the null context subjects may disagree [1]. For example in (ex. 3)

(ex. 3) She discussed her daughter's difficulties with the teachers.

In their study, 35% of subjects attached the PP to the VP and 65% to the NP.

More recently, evidence has been provided that argumenthood plays a part in the decision [13]. This suggests that all things being equal, the preferred interpretation is that a PP attaches to the verb as an argument.

This brief review indicates that even in psycholinguistic research, PP-attachment ambiguities remain an intriguing problem. No system will be able to resolve all such ambiguities given just information from the sentence. However, a wide range of information may help.

2.3 Ambiguity Resolving Systems

Many early computational systems used heuristics, or parsing principles, to resolve parsing ambiguities. One such principle is Right Association [7], which states that constituents should be attached to the nearest possible item. For PP attachment, this means that the PP should be attached to the nearest item, the NP.

On the other hand the minimal attachment heuristic suggests that the shallowest tree is built [2]. Following this principle, the PP would always attach to the verb.

Another commonly used and simple heuristic is to choose based on the preposition [5]. This is quite simple, and in the case of the preposition *of* is entirely effective; that is, a PP with the preposition *of* always attaches to the NP.

Another step forward is to choose based on the preposition, verb, and noun [4]. This is just one step away from the more commonly and more recently used four-tuple model.

The question still remains, what information should be used from the three or four-tuple? Machine learning algorithms have been used to derive information to use to resolve the attachment decision. These are corpus based, and a wide range of algorithms are used.

Ratnaparkhi proposed a maximum entropy model that used lexical information within verb phrases obtained from the Penn Treebank WSJ corpus and no external semantic knowledge [12]. They trained a maximum entropy model and a binary hierarchy of word classes derived by mutual information clustering from the corpus obtaining a resolution accuracy of 81.6%.

Stetina's corpus based model used decision trees and WordNet as a semantic dictionary to disambiguate word senses and resolve the PP-attachment ambiguity [14]. It attained an accuracy of 88.1%.

More recently, Nakov proposed a method that exploited the web as a very large training dataset, extracting its surface features and paraphrases based on the assumption that phrases found on the WWW are sometimes disambiguated and annotated by content creators [10]. Using the Ratnaparkhi dataset, they obtained an accuracy of 83.82% using n-gram models with statistics obtained by querying exact phrases including inflections and all possible variations of words derived from WordNet against WWW search engines.

Toutanova applied a random walk model to the task of PP-attachment attaining a resolution accuracy of 87.5%. Their supervised method used a Markov chain model to estimate word dependency distributions using WordNet synsets and stationary distribution used for making attachment predictions [15].

Clearly, a range of machine learning techniques have been used. However, none have made full use of hierarchical information in resolving the ambiguity. It is clear, that hierarchy is a key factor in human reasoning and language processing.

2.4 Semantic Nets and Hierarchies

The semantic net is a knowledge representation scheme that represents information in the form of graphs [3]. Vertices represent concepts and edges represent their relationships [11]. An example of a large semantic net is WordNet (see section 3.2). WordNet is a large semantic network with synonyms of words grouped into sets called synsets to form the basic building blocks (vertices).

Data in WordNet is grouped into word sense hierarchies. Word sense hierarchies are lexical trees formed by a sequence of hypernyms in different levels, with each level being trailed by the synset of its superordinate term. (In semantic net terminology if x is a hypernym of y, $yISAx$; here y might be instantiated by *dog* and x by *mammal*.) Figure 3 shows an example sense hierarchy of the noun *telescope*.

Word: telescope (noun)

```
scope
  => magnifier
     => ...
        => entity
```

Figure 3. Example sense hierarchy of the noun *telescope*

In WordNet, words, such as *see*, may have many senses, and thus may belong to many synsets. In this paper, this aspect is ignored; if more than one sense is present, the algorithm uses the first.

3 Algorithm

The system needs to decide the correct attachment for a PP given the input four-tuple. There are two stages. The first is supervised learning from a corpus of syntactically ambiguous examples where the correct attachment is known. The result of this stage is a set of lattices of known parsing results (see section 3.1) compiled from WordNet and the Penn Treebank (see section 3.2). The second stage is to select an attachment for a PP given the lattices (see section 3.3). All software is available on http://www.cwa.mdx.ac.uk/CABot/symbolicPP/PPattach.html.

3.1 Lattices

The algorithm builds a lattice [16]. For every four-tuple in the dataset, the sense hierarchies of the three elements (verb, NP-noun and PP-noun) are combined. Each vertex of the lattice is a triplet composed of one word from the sense hierarchies of each of the elements in the quadruple. The lattice contains every possible triplet combination of the three hierarchies. Figure 4 shows a small portion the lattice of the four-tuple *see girl with telescope*. Figure 4 uses the portion of the word *telescope* described above: $telescope, scope \rightarrow magnifier \rightarrow entity$; the *saw* hierarchy is represented by $saw \rightarrow perceive$; and *girl* is represented by $girl \rightarrow person \rightarrow entity$. The training lattices could all be combined into one lattice, but they were stored individually due to computer memory limitations.

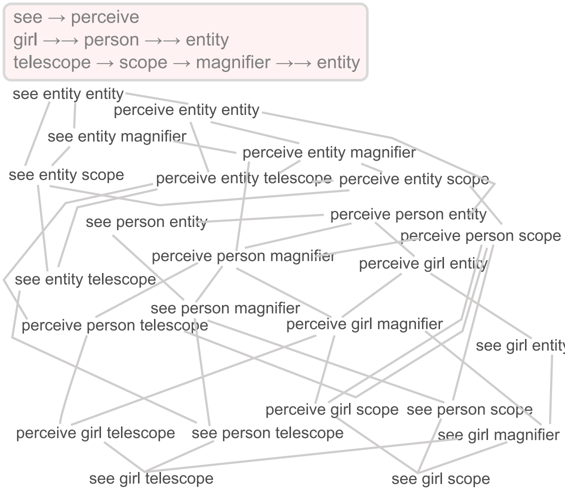

Figure 4. Lattice of see-girl-with-telescope

Assume the *verb* is represented by a hierarchy of i hierarchical levels with all the levels collectively containing V_i words, the NP *noun* consisting of j levels of N_j words and the PP *noun* consisting of k levels with P_k words. The size of the lattice is: $(\sum_{x=1}^{i} V_x * \sum_{y=1}^{j} N_y * \sum_{z=1}^{k} P_x)$.

In Figure 4 $i = 2$, $V_1 = 1$, $V_2 = 1$,$j = 3$, $N_1 = 1$, $N_2 = 1$, $N_3 = 1$, $k = 3$, $P_1 = 2$, $P_2 = 1$, $P_3 = 1$. This leaves two verbs, three nouns from the NP, and four nouns from the PP. So figure 4 has 24 nodes.

3.2 Datasets

Data for the simulation was drawn from two standard datasets. The first is WordNet, a large semantic lexicon for the English language comprised of information grouped into sets of synonyms or synsets [9]. WordNet is freely available to download, and is a widely used linguistic resource.

The corpus that was used was the Penn Treebank [8]. Testing and training data was based on the Wall Street Journal portion of the Penn Treebank corpora. The corpus has a standard annotated tree structure manually annotated by lexicographers consisting of parts-of-speech tags and the syntactic tree for each sentence [8]. Unfortunately, this is not freely available. None the less, the software for extracting the sentences, and converting those sentences into tuples is available on http://www.cwa.mdx.ac.uk/CABot/symbolicPP/PPattach.html.

3.3 Deciding From the Lattice

The algorithm predicts the attachment decision of a four-tuple by comparing its lattice against all the training lattices. The similarity measure of each pair of lattices is obtained by simply counting the number of intersecting vertices in them. This is illustrated in Figure 5. Training lattices that are derived from verb attachments contribute to the verb sum, and those with noun attachments contribute to the noun sum. If the verb sum is greater than the noun sum, a verb attachment is selected. Otherwise, a noun attachment is selected. If the noun and verb sums are equal, a verb attachment is predicted as occurrence of verb attachments have been observed to be more frequent than noun attachments.

Figure 5 shows the intersection of two four-tuples, *see girl with telescope* and *see boy with telescope*. Intersecting vertices are shown in dark.

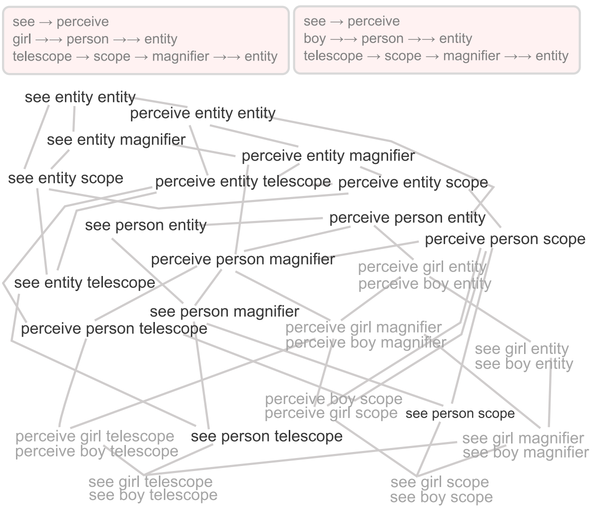

Figure 5. Intersection of lattices

4 Results

A total of **10694** sentences with PP ambiguities were extracted from the Wall Street Journal corpus of the Penn Treebank. It is known that all the ambiguities containing the preposition *of* are resolved to attach to the NP. So, the system was tested without these, and with these. Table 1 summarizes both of these tests.

The sentences are converted to four-tuples. In the first test, all four-tuples with the preposition *of* are removed. This leaves 7810 sentences. Of these, the first **3000** are retained for testing, with the remaining **4810** are used for training. The first two columns of data in table 1 refer to these results. Of the 3000 test sentences, 2415 resolve to VP attachments and 585 to NP attachments. The system performs better on VP attachments than NP attachments, and the total performance is 87.23%.

The second test included the four-tuples with *of*. Again, the first 3000 sentences were retained for testing, with the remainder used for training. However, the four-tuples with *of* were not included in the training set. During test, those with *of* were automatically assigned to be attached to the NP. This still leaves more training data, but not significantly more. The results of this test are in the final two columns of the table. The overall result was 90.53%.

Table 1. Test results

Attach.	Pred. no of	Acc. no of	Pred. of	Acc. of
verb	2224 / 2415	92.09%	1606 / 1741	92.23 %
noun	393 / 585	67.17%	1110 / 1259	88.17%
total	2617 / 3000	87.23%	2716 / 3000	90.53%

5 Discussion

Compared to other resolution systems, our method attained a comparatively high prediction accuracy. Unfortunately, there is no standard test corpora. Several systems [12, 17, 10] have used the Ratnaparkhi dataset, a subset of the Penn Treebank. Unfortunately, it is not clear to the authors what subset. So, it is difficult to conduct a fair and accurate comparison. Table 2 presents a comparison to previous work.

Note that the results include the *of* attachments. This seems to be the case with all other systems.

Table 2. Comparison with previous work

Method	Result
Maximum Entropy Model [12]	81.6%
Nearest neighbour method [17]	86.5%
Corpus based PP attachment ambiguity resolution with a semantic dictionary [10]	83.82%
Learning random walk models for inducing word dependency distributions [15]	87.5%
Decision trees and WordNet [14]	88.1%
Semantic hierarchies for lattice construction	90.5%

Table 2 is not a direct comparison between systems because the systems have been trained and tested on different corpora. While the Ratnaparkhi data is similar, it is still different. None the less, the system described in this paper performs better than all of the other systems. Moreover, the algorithm is a relatively simple one based on the combination of hierarchies.

The overall problem of PP attachment ambiguity resolution cannot be solved by four tuples, and indeed is not solvable in general. In the corpora, additional information is present within the sentence beyond that of the four items that are used in this paper. This information can influence the outcome of an attachment decision. For example, the introduction of *broken* in (ex. 4) changes the attachment decision.

(ex. 4) I saw the girl with the broken telescope.

Moreover, context beyond the sentence can also influence a decision. If the prior sentence is (ex. 5), the attachment of (ex. 1) is reversed.

(ex. 5) There was a girl with a telescope and one with a bat.

Finally, the problem is not generally solvable because different people make different decisions on the same sentence in the same context [1]. Consequently, no mere algorithm for attachment will ever be 100% correct, particularly if it is based only on a four-tuple. People some times make mistakes on attachment decisions.

Of course performance would be improved by a larger training set. In this paper, a lattice of hierarchies has been used to compensate for the sparseness of data. More data will almost certainly improve performance.

6 Conclusion

This paper is an illustration of the power of hierarchy. While much work have been done on resolving PP attachment ambiguity, none have so far used lattices of hierarchies. Complex statistical techniques have been used, but the simple use of hierarchies has surpassed them.

This paper has also shown that large corpora, such as the Penn Treebank and WordNet, are incredibly useful for developing systems for processing language. This is not new, but is another example of their benefits.

Acknowledgements

This work was supported by EPSRC grant EP/D059720.

References

[1] M. Ford, J. Bresnan, and R. Kaplan. A competence-based theory of syntactic closure. In J. Bresnan, editor, *The mental representation of grammatical relations*. MIT Press, 1982.

[2] L. Frazier. Processing sentence structure. In K. Raynoer, editor, *Eye Movements in Reading*. New York, NY: Academic Press, 1983.

[3] F. Harary. *Graph Theory*. Westview Press, 1994.

[4] D. Hindle and M. Rooth. Structural ambiguity and lexical relations. *Computational Linguistics*, 19:1:103–120, 1993.

[5] C. Huyck. A practical system for human-like parsing. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence*, 2000.

[6] R. Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, 2002.

[7] J. Kimball. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47, 1973.

[8] M. Marcus, B. Santorini, and M Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:2:313–330, 1993.

[9] G. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[10] P. Nakov and M. Hearst. Using the web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of HLT-NAACL*, 2005.

[11] M. Quillian. Word concepts: A theory of simulation of some basic semantic capabilities. *Behavioral Science*, 12:410–30, 1967.

[12] A. Ratnaparkhi, J. Reynar, and S. Roukos. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 250–252, 1994.

[13] C. Schutze and E. Gibson. Argumenthood and english prepostional phrase attachment. *Journal of Memory and Language*, 40:409–431, 1999.

[14] Jiri Stetina and Makoto Nagao. Corpus based pp attachment ambiguity resolution with a semantic dictionary. In *In Proceedings of WVLC*, pages 66–80, 1997.

[15] Kristina Toutanova, Andrew Ng Y., and Christopher Manning. Learning random walk models for inducing word dependency distributions. In *In Proceedings of ICML*, 2004.

[16] R. Wille. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23:493–515, 1992.

[17] S. Zhao and D. Lin. Nearest-neighbor method for resolving pp-attachment ambiguity. In *In Proceedings of IJCNLP-04*, pages 428–434, 2004.