A Comparison of Machine Learning Algorithms for Prepositional Phrase Attachment

Brian Mitchell & Robert Gaizauskas

Department of Computer Science, University of Sheffield,
Regent Court, Portobello Road, Sheffield. S1 4DP UK
{brianm,robertg}@dcs.shef.ac.uk

Abstract

This paper presents work which extends previous corpus-based work on training Machine Learning Algorithms to perform Prepositional Phrase attachment. Besides recreating others' experiments to see how algorithms' performance changes with the number of training examples and using *n*-fold cross-validation to produce more accurate error rates, we implemented our own vanilla Machine Learning Algorithms as a comparison. We also had people perform exactly the same task as the Machine Learning Algorithms to indicate whether the way forward lies in improving Machine Learning Algorithms or in improving the data sets used to train Machine Learning Algorithms. The results from all these experiments feed into our other work transforming the Penn TreeBank into a more useful resource for training Machine Learning Algorithms to do Prepositional Phrase attachment.

1. Introduction

This paper presents work which takes extra steps beyond previous attempts to train a learning algorithm for Prepositional Phrase Attachment using the Penn TreeBank (Marcus et al., 1993) as the data source. Besides recreating others' experiments — Error-Driven Transformation-Based Learning by Brill and Resnik (1994) — we extended these experiments in four ways: first by changing the number of training examples and second by performing 10-fold cross-validation. Moreover, we performed the same two comparisons with basic implementations of simple Machine Learning Algorithms (MLAs). Finally, a novel Web-based experiment to measure human performance at exactly the same task was set up.

Each of these extensions had a purpose: whereas the point of changing the number of training examples seen by a Machine Learning Algorithm was to monitor the attachment accuracy as this number increased, the 10-fold cross-validation experiments were designed to yield a more accurate picture of mean performance. The motivation for implementing two of the simplest Machine Learning Algorithms — a Naïve Bayesian Classifier and two variations of Decision Tree — was two-fold: to provide some real baselines against which to measure other's more sophisticated approaches and to check claims, for example (Brill, 1993, p38*f.*), that these simple algorithms are less suitable for PP attachment than their more sophisticated counterparts. The purpose of the Web-based human test was also two-fold: first to ratify some "*average human figures*" produced by Ratnaparkhi et al. (1994) but more importantly to suggest whether more mileage might be gained from a training set with more features rather than inching minor accuracy increases by developing Machine Learning Algorithms themselves.

2. The Task

The task investigated here involves making a binary decision about attaching a Prepositional Phrase (PP) to either a noun or verb. These attachment decisions were identified long ago and are commonly known as:

- Right Association — where a constituent tends to attach to another constituent immediately to its right: this favours attachment to the noun (Kimball, 1973)
- Minimal Attachment — where a constituent tends to attach to an existing nonterminal using the fewest additional syntactic nodes: this favours attachment to the verb (Frazier, 1978)

However work by Whittemore et al. (1990) showed that neither Right Association nor Minimal Attachment account for more than 55% of cases — the actual attachment ratio depends on the corpus — and work by Taraban and McClelland (1990) showed that these structural models are poor predictors of people's behaviour when resolving ambiguity. Both these works found lexical preferences to be the key to resolving attachment ambiguity. Based on these premises, Hindle and Rooth (1993) in their landmark research decided to use the co-occurrence of verbs and nouns with specific prepositions in 13,000,000 words of the Associated Press Corpus as an indicator of "*Lexical Association*." From then on, corpus-based training for automatic PP attachment has been dominated by methods that utilise some form of Mutual Information generated by the co-occurrence of vocabulary.

For their research, Brill and Resnik (1994) used the Penn TreeBank (PTB) utility `tgrep` to extract 12,766 quadtuples from the near 165,000 sentences[1] of Wall Street Journal (WSJ) text in the PTB corpus, reserving 500 of these examples for testing. Note that these quadtuples are only for sentences matching the pattern (v × n1 × p × n2) for example:

see/v the boy/n1 on/p the hill/n2

Such sentences account for 7·75% of the corpus and are inherently structurally ambiguous, though not always semantically ambiguous.

For the original research by Brill & Resnik, the vocabulary in both the training and test sets was reduced to its root

[1] The PTB2 has 164,798 items marked as sentences, including 49,208 marked as top-level sentences, leaving 115,590 embedded sentences, such as reported speech.

form, to maximise the scope of vocabulary. So an actual example reads:

ban, *trade*, *through*, *computer*
when the original sentence was:

> In Washington, House aides said Mr. Phelan told congressmen that the collar, which *banned* program *trades through* the Big Board's *computer* when the Dow Jones Industrial Average moved 50 points, didn't work well.
>
> *Penn TreeBank file* `wsj_0088`

The examples are also presented in this way for the manual experiment, making the human task identical to that of the Machine Learning Algorithm used by Brill & Resnik.

3. Baselines

The random chance of correct attachment reflects the distribution of attachments in a given data set, which for the PTB is approximately 64:36 for nominal:verbal attachment distribution. So attaching all PPs to the noun (Right Attachment) gives 64% attachment accuracy as a baseline — this is reasonably close to the general 55% figure offered by Whittemore et al. This 64% figure was confirmed using the `ZeroR` learning scheme from `Weka`.[2] `ZeroR` simply predicts the majority class in the training data, nominal attachment in this case, but as part of its execution the scheme outputs various statistics including the distribution of examples between classifications: 63·4% in this case.

In their experiments, Brill & Resnik achieved 80·8% accuracy using the words alone (though they did manage 81·1% by adding word-class information). Others have reached similar scores on the same task using the same corpus: Stetina and Nagao (1997) also used semantic information to attain 88% accuracy (the highest score yet and done with a decision tree); Collins and Brooks (1995) used Backed-Off Estimation and scored 84·1% without morphological processing and 84·5% with it; and Ratnaparkhi et al. (1994), who used Maximum Entropy, scored 77·7% using just the words and 81·6% by adding word-class information. That paper also produced two sets of "*average human figures*" (where "*average*" applied to the figures, not the humans) by taking the average scores of three treebanking experts (presumably the three authors) tested on three hundred sentences selected randomly from the corpus. Two human figures were produced: 88·2% accuracy using just the four head words (*á la* Brill & Resnik) and 93·2% using the whole sentence. Interestingly, the human scores from Ratnaparkhi et al. are not borne out by our own experiments, see §5.

4. Recreating the PTB2 Experiments

As already mentioned, numerous Machine Learning Algorithms have been applied to the problem of Prepositional Phrase attachment. However, the algorithms that have been most famously reported are not the simplest Machine Learning Algorithms available. Therefore, we elected to implement our own vanilla versions of two of the simplest Machine Learning Algorithms: a Naïve Bayesian Classifier and a Decision Tree, all based on the descriptions in Mitchell (1997). Although the mathematics behind EDTBL is simpler than some implementations of Decision Trees that utilise pruning, our implementations were deliberately kept as simple as possible: having no pruning and using a basic version of the `ID3` algorithm to partition the data.

We created two versions of the Decision Tree: one using the standard Information Gain (IG) metric based on entropy to rank the contribution of each attribute per example, the other using Gain Ratio (GR) which is an extension of IG that penalises discriminants chosen from large sets. This means that GR tries to avoid deciding the attachment based on the actual vocabulary of nouns and verbs, preferring to make decisions using the preposition where possible.

The availability of the original data and software used by Brill and Resnik (1994) not only enabled the recreation of their experiment — gratifyingly obtaining exactly the same accuracy — but also enabled a series of other experiments using our own software on a data set which, although not exactly a universal standard, had already been tested by experts and which is available to anyone via the Internet.

4.1. Phase 1

For this phase of the experiments, the original training and test data from Brill and Resnik (1994) were used and kept in their original ordering, although it was necessary to change the structure of the data files to allow the other learning algorithms to read them. A particular Machine Learning Algorithm was selected, trained, and always evaluated on the same test data of 500 examples. To measure how quickly the Machine Learning Algorithm in question became competent at Prepositional Phrase attachment, the number of training examples was gradually increased from a small number (just 100 examples) until all of the 12,266 examples were seen by the learning phase. The selection of the training examples was always a contiguous sample of the complete training data, always starting from the beginning. The results of all the algorithms run in Phase 1 appear in Figure 1.

More results with smaller amounts of training data (fewer than 1,000) were obtained deliberately, partly because we were curious by how much the accuracy would fluctuate with so few training examples, and partly because we wanted to compare these algorithms' performance with that of a Support Vector Machine (SVM). The SVM[3] is limited to utilising fewer than 2,000 training examples: this limitation applies to SVMs in general because of they way they manipulate the data. As a check, we also used the SVM supplied with `Weka` and hit similar limitation and performance figures. Both SVMs consistently produced 73% to 77% accuracy even with as few as 200 training examples. The SVMs' figures are not shown in Figure 1 because they would be difficult to distinguish in the crowded left-hand side of the graph.

[2] The *Waikato Environment for Knowledge Analysis*, a `Java` software workbench from the University of Waikato that is freely available from `http://www.waikato.ac.nz/~ml/weka`

[3] The software was written by Vincent Wan of the *Speech and Hearing Group* at Sheffield University

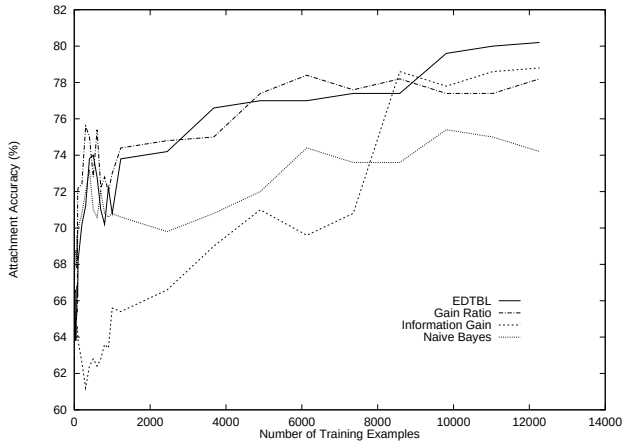

Figure 1: Results from Phase 1

4.2. Discussion of Phase 1 Results

The most striking event shown by Figure 1 is the rapid jump up by the Decision Tree using Information Gain as part of its `ID3` algorithm. One advantage of the Decision Tree method is that its tree of rules is meaningful to a human reader and a manual inspection of the trees at either end of the jump showed that the Decision Tree had suddenly "*realised*" that using the preposition itself as the main discriminant rather than the other vocabulary (the nouns and verbs) had more benefit. Further analysis of the region between 7,000 and 8,500 training examples (using the same data in different orders) showed that the switch from nouns and verbs to the preposition as the main discriminant was always quite sudden and always occurring around the 8,000 mark. The conclusion from this is that using `ID3` with Information Gain is less beneficial than using it with Gain Ratio if the system has insufficient training data to realise that prepositions have more worth as the primary discriminant. But once the Information Gain Decision Tree has seen sufficient training data, as shown in the results of Phase 2, it produces similar accuracy scores to Gain Ratio.

A second interesting feature shown by Figure 1 is that the Naïve Bayesian Classifier is the only algorithm that ends on a downward trend. Of course, were more training data available, the Naïve Bayesian Classifier may well improve again, even if the improvement were merely the next phase of its apparently oscillating accuracy.

A third interesting features of Figure 1 is the relative performance of EDTBL and the Decision Tree using Gain Ratio. Between about 4,200 and 9,000 training examples, Gain Ratio actually performs better, although EDTBL finishes with a 2% absolute advantage and a relative error difference of 9·17%. The continuation of this contest between Decision Trees and EDTBL becomes even more interesting in Phase 2.

4.3. Phase 2

Whereas Phase 1 already had pre-prepared sets of training and test data, Phase 2 needed to reorganise those data into *n*-fold cross-validation data, in this case ten sets. First the training and test data were merged into a single set, then the order randomised. This randomly ordered set was then used to create ten cross-validation sets, achieved by splitting the entire data set into ten parts. Each tenth became test data, one for each of the ten sets, with the remaining nine-tenths of the data forming the training data for that set. The order of the data in each training set was then randomised again. Any significant gains or losses in algorithm performance due to data ordering are rendered unlikely due to this double randomisation and the *n*-fold cross-validation.

Each of the chosen Machine Learning Algorithms was run in turn over the ten sets of cross-validation data. For each cross-validation set, the number of training examples was increased, as in Phase 1, until it covered all 11,490 training examples (stepped data). Whereas Phase 1 used 500 test cases (4·08% of the data), the 10-fold cross-validation uses 10% of the data, or 1,276 test cases. The mean percentage of attachment accuracy and its standard deviation were computed for the same data step over all ten of the cross-validation sets. These figures for each Machine Learning Algorithm were merged into the same graph for comparison purposes (see Figure 2), though the standard deviation only appears in the individual graphs (Figures 3, 4, and 5) for clarity.

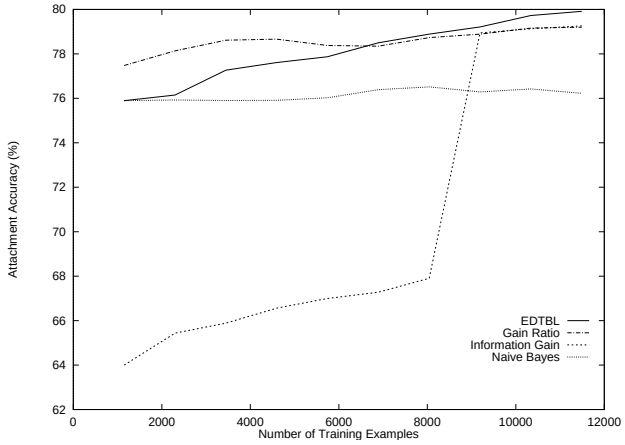

Figure 2: Results from Phase 2

4.4. Discussion of Phase 2 Results

As in Figure 1, the most striking event is the rapid jump up (always in approximately the same place in each set of the *n*-fold cross-validation) by the Decision Tree using Information Gain. Once the Information Gain Decision Tree has made this jump, it performs with a mean accuracy quite consistent with that of the Decision Tree using Gain Ratio. Before that point, however, the performance of the two algorithmic variations is staggeringly different, not only in absolute terms with GR massively outperforming IG, but also in behaviour: whereas Information Gain is always improving, Gain Ratio wavers around 78%. Should an application require a fast and simple algorithm that works well having seen only a few thousand training examples, then a Decision Tree using Gain Ratio is worth considering.

The oscillating accuracy of the Naïve Bayesian Classifier suggested by Figure 1 can now be seen clearly. Whilst there is the possibility that with a vast amount of training data, perhaps measured in millions, such a classifier may prove a useful algorithm since it trains and classifies

rapidly. For the amount of training data currently available, however, it is a poor choice, though it does do somewhat better than a random guess.

It is now clear that it takes about 6,500 training examples for EDTBL to overhaul the Decision Tree using Gain Ratio. But once it starts performing better, it stays performing better: EDTBL is ultimately about 1% more accurate than the Decision Trees in both absolute and relative terms. However, a glance at Figures 3, 4, and 5 shows that by taking standard deviation into account, the Decision Trees can outperform EDTBL when trained on 11,490 examples.

The 10-fold cross-validation experiments demonstrate that Error-Driven Transformation-Based Learning is not outstandingly better than `ID3`, thus refuting Brill's implication in his thesis that Decision Trees are less suitable than EDTBL for Prepositional Phrase attachment. Although the experiments also show that EDTBL does not need many examples to perform quite well, the same can be said for Decision Tree using `ID3` and Gain Ratio.

An extra result, not shown in any of the graphs, came about whilst using `Weka`'s `ZeroR` classifier to validate the baseline in preparation for this paper: we also tried their implementation of a Decision Table. Using the same training and test data as in Phase 1, we obtained an accuracy of 84·56%. Unfortunately, it was not possible to produce a comprehensive set of Phase 1 and Phase 2 results in time.

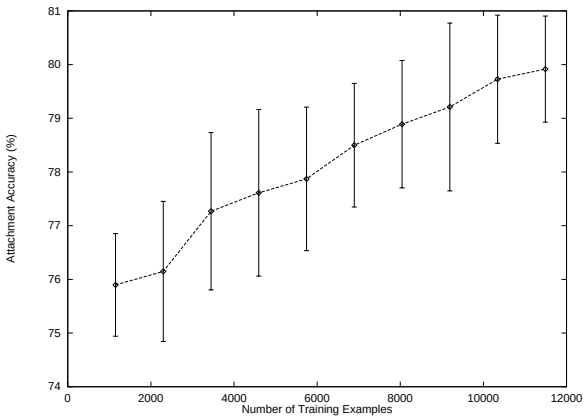

Figure 3: EDTBL Results from Phase 2

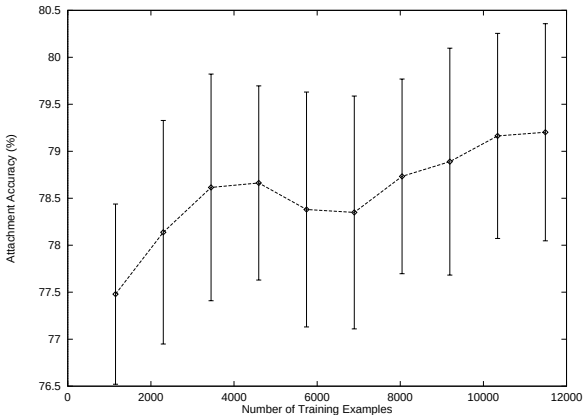

Figure 4: `ID3` Gain Ratio Results from Phase 2

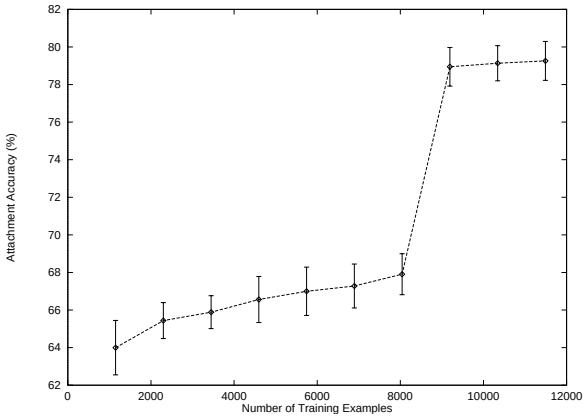

Figure 5: `ID3` Information Gain Results from Phase 2

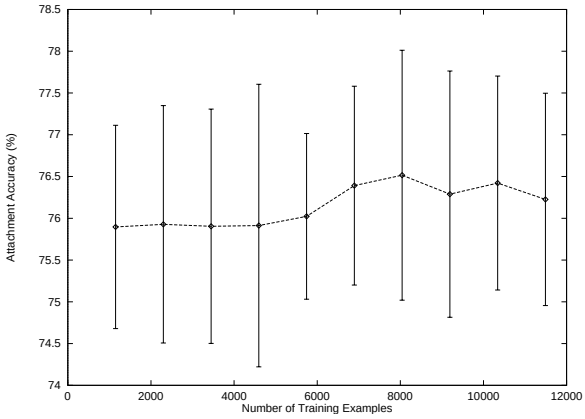

Figure 6: Naïve Bayesian Classifier Results from Phase 2

5. Manual PP Attachment Experiments

All the techniques mentioned previously generally obtain similar scores and all perform less well than we would like. So is the problem with the algorithms themselves or with the representation of the data? One way to decide is to see whether people can outperform the computer on a level playing field. So we created some software to test humans on the task exactly as it was given to the Machine Learning Algorithms. The data shown to the human subjects are the same as those shown to the MLAs; indeed the data are that of the test set used in the experiments described above and by Brill & Resnik. Besides working with the data, human users also have the opportunity to mark any examples that they believe are genuinely ambiguous.

Our software, written in `Java` and supported by `CGI` scripts, has been implemented to run in a browser over the World Wide Web, thus making the experiment accessible to as large an audience as possible.[4] The main part of the software displays ten pages of fifty examples each, where each example contains five words: the verb, the noun, the preposition, the head noun of the PP's complement, and the word `ambiguous`. The first two fields and `ambiguous` are clickable and become highlighted with a tick when selected. Checking `ambiguous` is how users indicate they

[4]The experiments are still on-line and can be accessed via `http://www.dcs.shef.ac.uk/~brianm`

believe a particular example to be genuinely ambiguous. The selection of a noun or verb within a particular example is mutually exclusive: selecting one deselects the other if it had already been selected. Note that even though a user may believe an example to be ambiguous, a decision must still be made between nominal or verbal attachment. Besides presenting the attachment examples, the software also requests some personal details which includes whether or not a user is a native speaker of English and what language or grammar expertise that person has. These details allow a quaternary breakdown of results: native speakers versus non-native speakers, orthogonal to experts versus non-experts.

5.1. Discussion of Results and Issues from the Online Manual Experiment

Although the experiment is still very much underway, there have already been some interesting results and feedback.

In terms of results, our "*average human figures*" are significantly lower than those of Ratnaparkhi et al. Whereas their three treebanking experts obtained 88·2% on the same kind of test as ours (though on 300 rather than 500 examples), the three language experts who have used our software so far have averaged 75·7%. Two non-native speakers have so far completed all the examples, one scoring the lowest mark to date, 66·4%, the other the highest, 78·8%. The software only came on-line a few days before the deadline, so there were only five completed answers available to report here but more than a dozen people have started completing answers. But the fact that no-one has even come close to 88% suggests that the treebankers' particular expertise gave them a huge advantage. The eventual hope for our "*average human figures*" is that "*average*" will apply not only to the figures themselves (mean scores) but also to the humans: the mean score of an average (not particularly expert) user. So far this figure is 73·4%.

There has been some interesting feedback from participants. Every one found the exercise difficult. Those with some language expertise said they had no trouble understanding the task itself, though they still found some of the attachment decisions hard to mark. The non-experts found both the concept of the task and the task itself difficult. They said they would have liked to have seen more examples before embarking on the exercise. Presumably all the experts had already heard of the problem of Prepositional Phrase attachment and so had a better understanding of the task. So far, two people who started the task have decided not to finish because they found it too taxing.

Allowing the user to choose `ambiguous` seems to have caused some confusion, though not in the way ambiguity in Prepositional Phrase attachment usually does. Some people seem to have selected `ambiguous` for those examples that they found more difficult, in particular if they found it hard to imagine a sentence containing the four head words, rather than because the attachment really is ambiguous. Others have disputed the worth of the `ambiguous` field itself, arguing that if an example is ambiguous, then the attachment either cannot be decided or does not matter since it will not change the meaning of the sentence. The intention behind putting the `ambiguous` choice in was to obtain a rough frequency count of genuinely ambiguous attachments. It was also intended to see whether any prepositions, for example "*in*" or "*on*", or kinds of preposition, such as locative or temporal, cause more ambiguity than others to humans. In the end, the data from this field may instead yield a picture of the kinds of example that people find more difficult, whether or not they are actually ambiguous. Such examples may also vary between native and non-native speakers and between language experts and non-experts.

Of course by scoring people's efforts, there is an implication that humans can actually achieve 100% accuracy with Prepositional Phrase attachment. But we acknowledge that the answers obtained from the PTB may contain errors, therefore further experimentation may help: for example a similar web-based experiment showing the entire sentence not just the head words. The inter-annotator agreement in such an experiment could help define a more acceptable set of answers, should there be any discrepancies with the answers defined in the PTB. Of course, there may be cases of genuine ambiguity where changing the attachment of the PP makes no difference to the meaning of the sentence, in which case the inter-annotator agreement may well be low: an interesting fact in its own right.

6. Discussion

The comparison between EDTBL and the simpler Machine Learning Algorithms is interesting. Despite Brill dismissing Decision Trees in his thesis, they actually do respectably well, being within 1% of EDTBL and 2% of Maximum Entropy but still a few percent behind Backed-Off Estimation. Initially, using the vocabulary as the main discriminant yields poor results for the Decision Tree using Information Gain but once it has seen about 8,000 examples, it suddenly "*realises*" that the preposition is the single best classifier to determine attachment and there is a consequentially staggering jump in accuracy. Yet unlike Gain Ratio which starts out using the preposition as the primary discriminant and which is permanently penalised for using the nouns and verbs, Information Gain retains the flexibility to discriminate using actual nouns and verbs if need be. There is an as yet unconfirmed possibility from the `Weka` experiment, that a Decision Table may be even better than a Decision Tree for the Prepositional Phrase attachment task. In comparison, the experiments have shown that the Naïve Bayesian Classifier performs poorly and seems to be a substantially inferior choice of algorithm for PP attachment.

The closeness at the top end of the scale between several different Machine Learning Algorithms and the bounds set by the average human figures for the head word tests, suggests that existing algorithms are nearly attaining the best realistic scores with the available data. However, the human results from Ratnaparkhi et al. (1994) based on attachment using the full sentence rather than just the head words, indicate that Machine Learning Algorithms could do better given a more finely featured set of attributes on which to train.

7. Further Work

Prior to the experiments described in this paper, we reworked the Penn TreeBank into a new form to make it easier to extract a richer feature set for Prepositional Phrase attachment. This was triggered by the realisation that the standard binary attachment decision between noun and verb is not the only pattern in which Prepositional Phrases occur ambiguously. This observation has since been borne out by Fang (2000) both independently of our research and on a different corpus, ICE-GB (Greenbaum, 1992; Greenbaum, 1996). Fang finds that the canonical (v × n1 × p × n2) quadtuple accounts for less than a quarter of ambiguous cases in the ICE corpus.

Our reworking of the PTB involves several transformations, including relabeling Noun Phrases as Simple Noun Phrases if they contain no embedded phrases, grouping verbs in the manner of a chunking parser to replace the highly nested right-branching structure inherent in the PTB, and adding explicit pointers to the head words of phrases, though this step is not without difficulty since the definition of a head word is disputed in the field. The hope is that by making the more information explicit, a richer set of attributes on which to train MLAs will become more evident. We believe that the basic four head words are simply too impoverished to allow Machine Learning Algorithms to approach human performance with full sentences.

8. References

Eric Brill and Philip Resnik. 1994. A Rule-Based Approach to Prepostional Phrase Attachment Disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING–94)*, Kyoto, Japan.

Eric Brill. 1993. *A Corpus-Based Approach To Language Learning*. Ph.D. thesis, University of Pennsylvania.

Michael Collins and James Brooks. 1995. Prepositional Phrase Attachment through a Backed-off Model. In David Yarowsky and Kenneth Church, editors, *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 27–38, Cambridge, Massachusetts, June.

Alex Chengyu Fang. 2000. A Lexicalist Approach towards the Automatic Determination for the Syntactic Functions of Prepositional Phrases. *Natural Language Engineering*, 6(2):183–201, June.

Lyn Frazier. 1978. *On Comprehending Sentences: Syntactic Parsing Strategies*. Ph.D. thesis, University of Connecticut.

S Greenbaum. 1992. A new Corpus of English: ICE. In J Svartvik, editor, *Directions in Corpus Linguistics: Proceedings of the Nobel Symposium 82*, pages 171–179, Stockholm, Sweden. Mouton de Gruyter, Berlin.

S Greenbaum, editor. 1996. *Comparing English Worldwide: The International Corpus of English*. Oxford University Press.

Donald Hindle and Mats Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1):103–120.

J. Kimball. 1973. Seven Principals of Surface Structure Parsing in Natural Language. *Cognition*, 2:15–47.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tom Mitchell. 1997. *Machine Learning*. McGraw-Hill, International edition.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey, March.

Jiri Stetina and Makoto Nagao. 1997. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In Jou Zhou and Kenneth Church, editors, *Proceedings of 5th Workshop on Very Large Corpora*, pages 66–80, Beijing and Hong Kong, August 18–20.

Roman Taraban and James L. McClelland. 1990. Constituent Attachment and Thematic Role Assignment in Sentence Processing: Influences of Content-Based Expectations. *Journal of Memory and Language*, 27:23–30.

Greg Whittemore, Kathleen Ferrara, and Hans Brunner. 1990. Empirical Study of Predictive Powers of Simple Attachment Schemes for Post-Modifier Prepositional Phrases. In *Proceedings of the 28th Annual Meeting of the Association for Computation Linguistics*, pages 23–30.

Acknowledgments

We would like to thank those people who have participated in our on-line experiments and Vincent Wan of the Department of Computer Science, Sheffield University, for helping set up the on-line experiments and providing us with a custom-written Support Vector Machine.