A Maximum Entropy Model for Prepositional Phrase Attachment

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos*

IBM Research Division
Thomas J. Watson Research Center
Yorktown Heights, NY 10598

1. Introduction

A parser for natural language must often choose between two or more equally grammatical parses for the same sentence. Often the correct parse can be determined from the lexical properties of certain key words or from the context in which the sentence occurs. For example in the sentence,

In July, the Environmental Protection Agency imposed a gradual ban *on virtually all uses of asbestos*.

the prepositional phrase *on virtually all uses of asbestos* can attach to either the noun phrase *a gradual ban* , yielding

[$_{VP}$ imposed [$_{NP}$ a gradual ban [$_{PP}$ on virtually all uses of asbestos]]],

or the verb phrase *imposed*, yielding

[$_{VP}$ imposed [$_{NP}$ a gradual ban] [$_{PP}$ on virtually all uses of asbestos]].

For this example, a human annotator's attachment decision, which for our purposes is the "correct" attachment, is to the noun phrase. We present in this paper methods for constructing statistical models for computing the probability of attachment decisions. These models could be then integrated into scoring the probability of an overall parse. We present our methods in the context of prepositional phrase (PP) attachment.

Earlier work [11] on PP-attachment for verb phrases (whether the PP attaches to the preceding noun phrase or to the verb phrase) used statistics on co-occurences of two bigrams: the main verb (V) and preposition (P) bigram and the main noun in the object noun phrase (N_1) and preposition bigram. In this paper, we explore the use of more features to help in modeling the distribution of the binary PP-attachment decision. We also describe a search procedure for selecting a "good" subset of features from a much larger pool of features for PP-attachment. Obviously, the feature search cannot be guaranteed to be optimal but appears experimentally to yield a good subset of features as judged by the accuracy rate in making the PP-attachment decisons. These search strategies can be applied to other attachment decisions.

We use data from two treebanks: the IBM-Lancaster Treebank of Computer Manuals and the University of Pennsylvania WSJ treebank. We extract the verb phrases which include PP phrases either attached to the verb or to an object noun phrase. Then our model assigns a probability to either of the possible attachments. We consider models of the exponential family that are derived using the Maximum Entropy Principle [1]. We begin by an overview of ME models, then we describe our feature selection method and a method for constructing a larger pool of features from an exisiting set, and then give some of our results and conclusions.

2. Maximum Entropy Modeling

The Maximum Entropy model [1] produces a probability distribution for the PP-attachment decision using only information from the verb phrase in which the attachment occurs. We denote the partially parsed verb phrase, i.e., the verb phrase without the attachment decision, as a history h, and the conditional probability of an attachment as $p(d|h)$, where $d \in \{0,1\}$ and corresponds to a noun or verb attachment (respectively). The probability model depends on certain features of the whole event (h,d) denoted by $f_i(h,d)$. An example of a binary-valued feature function is the indicator function that a particular (V,P) bigram occured along with the attachment decision being V, i.e. $f_{print,on}(h,d)$ is one if and only if the main verb of h is "print", the preposition is "on", and d is "V". As discussed in [6], the ME principle leads to a model for $p(d|h)$ which maximizes the training data log-likelihood,

$$\sum_{h,d} \tilde{p}(h,d)\ log\ p(d|h),$$

where $\tilde{p}(h,w)$ is the empirical distribution of the training set, and where $p(d|h)$ itself is an exponential model:

*Jeff Reynar, from University of Pennsylvania, worked on this project as a summer student at I.B.M.

$$p(d|h) = \frac{\prod_{i=0}^{k} e^{\lambda_i f_i(h,d)}}{\sum_{d=0}^{1} \prod_{i=0}^{k} e^{\lambda_i f_i(h,d)}}$$

At the maximum of the training data log-likelihood, the model has the property that its k parameters, namely the λ_i's, satisfy k constraints on the expected values of feature functions, where the ith constraint is,

$$E_m f_i = \tilde{E} f_i$$

The model expected value is,

$$E_m f_i = \sum_{h,d} \tilde{p}(h) p(d|h) f_i(h,d)$$

and the training data expected value, also called the desired value, is

$$\tilde{E} f_i = \sum_{h,d} \tilde{p}(h,d) f_i(h,d)$$

The values of these k parameters can be obtained by one of many iterative algorithms. For example, one can use the *Generalized Iterative Scaling* algorithm of Darroch and Ratcliff [3]. As one increases the number of features, the achievable maximum of the training data likelihood increases. We describe in Section 3 a method for determining a reliable set of features.

3. Features

Feature functions allow us to use informative characteristics of the training set in estimating $p(d|h)$. A feature is defined as follows:

$$f_i(h,d) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{iff } d = 0 \text{ and } \forall q \in \mathcal{Q}_i, q(h) = 1 \\ 0, & \text{otherwise.} \end{cases}$$

where $\mathcal{Q}_i$ is a set of binary-valued questions about h. We restrict the questions in any $\mathcal{Q}_i$ ask only about the following four *head* words:

1. Head Verb (V)
2. Head Noun (N_1)
3. Head Preposition (P)
4. Head Noun of the Object of the Preposition (N_2)

For example, questions on the history "imposed a gradual ban on virtually all uses of asbestos", can only ask about the following four words:

```
imposed ban on uses
```

The notion of a "head" word here corresponds loosely to the notion of a lexical head. We use a small set of rules, called a *Tree Head Table*, to obtain the head word of a constituent [12].

We allow two types of binary-valued questions:

1. Questions about the presence of any n-gram ($n \leq 4$) of the four head words, e.g., a bigram maybe $\{$`V == ''is'',P == ''of''`$\}$. Features comprised solely of questions on words are denoted as "word" features.
2. Questions that involve the class membership of a head word. we use a binary hierarchy of classes derived by *mutual information* clustering which we describe below. Given a binary class hierarchy, we can associate a bit string with every word in the vocabulary. Then, by querying the value of certain bit positions we can construct binary questions. For example, we can ask whether about a bit position for any of the four head words, e.g., `Bit 5 of Preposition == 1`. We discuss below a richer set of these questions. Features comprised solely of questions about class bits are denoted as "class" features, and features containing questions about both class bits and words are denoted as "mixed" features[1].

Before discussing, feature selection and construction, we give a brief overview of the mutual information clustering of words.

Mutual Information Bits Mutual information clustering, as described in [10], creates a a class "tree" for a given vocabulary. Initially, we take the C most frequent words (usually 1000) and assign each one to its own class. We then take the $(C+1)$st word, assign it to its own class, and merge the pair of classes that minimize the loss of average mutual information. This repeats until all the words in the vocabulary have been exhausted. We then take our C classes, and use the same algorithm to merge classes that minimize the loss of mutual information, until one class remains. If we trace the order in which words and classes are merged, we can form a binary tree whose leaves consists of words and whose root is the class which spans the entire vocabulary. Consequently, we uniquely identify each word by its path from the root, which

[1] See Table 7 for examples of features

can be represented by a string of binary digits. If a path lengt of a word is less than the maximum depth, we pad the bottor of the path with 0's (dummy left branches), so that all word are represented by an equally long bitstring. "Class" feature query the value of bits, and hence examine the path of th word in the mutual information tree.

Special Features In addition to the types of features de scribed above, we employ two special features in the Ml model, the *Complement* and the *Null* feature. The Comple ment, defined as

$$f_{comp}(h,d) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{iff } f_i(h,d) = 0, \forall f_i \in \mathcal{M} \\ 0, & \text{otherwise.} \end{cases}$$

will fire on a pair (h, d) when no other f_i in the model applie: The Initial feature is simply

$$f_{null}(h,d) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{iff } d = 0 \\ 0, & \text{otherwise} \end{cases}$$

and causes the ME model to match the `a priori` probability of seeing an N-attachment.

3.1. Feature Search

The search problem here is to find an optimal set of features $\mathcal{M}$ for use in the ME model. We begin with a search space $\mathcal{P}$ of putative features, and use a feature ranking criterion which incrementally selects the features in $\mathcal{M}$, and also incrementally expands the search space $\mathcal{P}$.

Initially $\mathcal{P}$ consists of all 1, 2, 3 and 4-gram word features of the four headwords that occur in the training histories[2], and all possible unigram class features[3]. We obtain $\sum_{k=1}^{4} \binom{4}{k} = 15$ word features from each training history, and, assuming each word is assigned m bits, a total of $2m * 4$ unigram class features, e.g., there are $2m$ features per word: `Bit 1 of Verb == 0, Bit 1 of Verb == 1, ... , Bit` m `of Verb == 0, Bit` m `of Verb ==1`

The feature search then proceeds as follows:

1. Initialize $\mathcal{P}$ as described above, initialize $\mathcal{M}$ to contain complement and null feature
2. Select the best feature from $\mathcal{P}$ using Delta-Likelihood rank
3. Add it to $\mathcal{M}$

[2] With a certain frequency cut-off, usually 3 to 5

[3] Also with a certain frequency cut-off

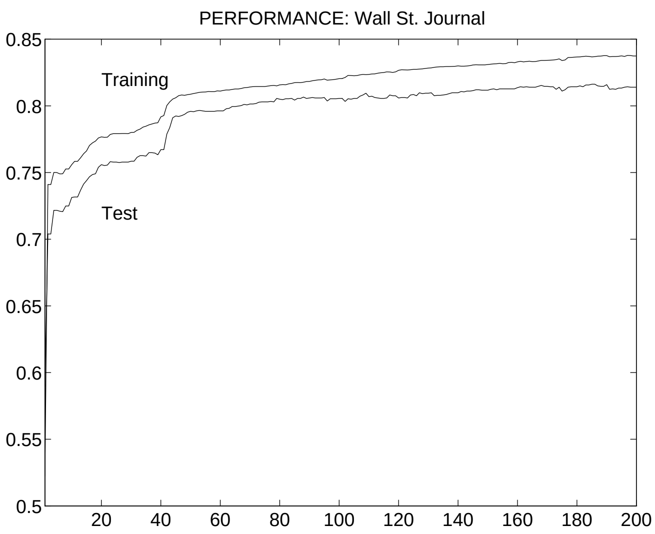

Figure 1: Performance of Maximum Entropy Model on Wall St. Journal Data

4. Train Maximum Entropy Model, using features in $\mathcal{M}$
5. Grow $\mathcal{P}$ based on last feature selected
6. repeat from (2)

If we measure the training entropy and test entropy after the addition of each feature, the training entropy will monotonically decrease while the test entropy will eventually reach a minimum (due to overtraining). Test set performance usually peaks at the test entropy minimum (see Fig. 1 & 2).

Delta-Likelihood At step (2) in the search, we rank all features in $\mathcal{P}$ by estimating their potential contribution to the log-likelihood of the training set. Let q be the conditional probability distribution of the model with the features currently in $\mathcal{M}$. Then for each $f_i \in \mathcal{P}$, we compute, by estimating only λ_i, the probability distribution p that results when f_i is added to the ME model:

$$p(d|h) = \frac{q(d|h)e^{\lambda_i f_i(h,d)}}{\sum_{w=0}^{1} q(w|h)e^{\lambda_i f_i(h,w)}}$$

We then compute the increase in (log) likelihood with the new model:

$$\delta L_i = \sum_{h,w} \tilde{p}(h,w) \ln p(w|h) - \sum_{h,w} \tilde{p}(h,w) \ln q(w|h)$$

and choose the feature with the highest δL. Features redundant or correlated to those features already in $\mathcal{M}$ will produce

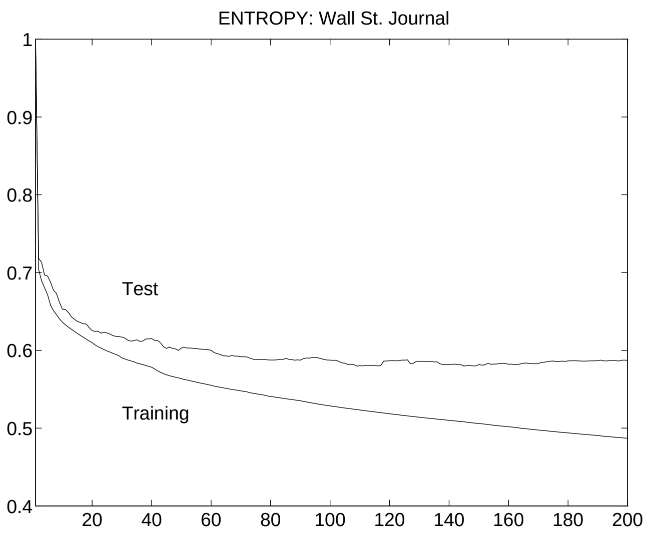

Figure 2: Entropy of Maximum Entropy Model on Wall St. Journal Data

a zero or negligible δL, and will therefore be outranked by genuinely informative features. The chosen feature is added to $\mathcal{M}$ and used in the ME Model.

3.2. Growth of Putative Feature Set

At step (5) in the search we expand the space $\mathcal{P}$ of putative features based on the feature last selected from $\mathcal{P}$ for addition to $\mathcal{M}$. Given an n-gram feature f_i (i.e., of type "word", "class" or "mixed") that was last added to $\mathcal{M}$, we create $2m*4$ new $n+1$-gram features which ask questions about class bits in addition to the questions asked in f_i. E.g., let $f_i(h,d)$ constrain $d = 0$ and constrain h with the questions `V == ''imposed'', P == ''on''`. Then, given $f_i(h,d)$, the $2m$ new features generated for just the Head Noun are the following:

```
V == ''imposed'', P == ''on'',
Bit 1 for Noun == 0

V == ''imposed'', P == ''on'',
Bit 1 for Noun == 1

        ⋮

V == ''imposed'', P == ''on'',
Bit m for Noun == 0

V == ''imposed'', P == ''on'',
Bit m for Noun == 1
```

We construct the remaining $6m$ features similarly from the remaining 3 head words. We skip the construction of features containing questions that are inconsistent or redundant with those word or class questions in f_i.

	Computer Manuals	Wall St. Journal
Training Events	8264	20801
Test Events	943	3097

Table 1: Size of Data

The newly created features are then added to $\mathcal{P}$, and compete for selection in the next Delta-Likelihood ranking process. This method allows the introduction of complex features on word classes while keeping the search space manageable; $\mathcal{P}$ grows linearly with $\mathcal{M}$.

4. Results

We applied the Maximum Entropy model to sentences from two corpora, the I.B.M. Computer Manuals Data, annotated by Univ. of Lancaster, and the Wall St. Journal Data, annotated by Univ. of Penn. The size of the training sets, test sets, and the results are shown in Tables 1 & 2.

The experiments in Table 2 differ in the following manner:

"Words Only" The search space $\mathcal{P}$ begins with all possible n-gram word features with n being 1, 2, 3,or 4; this feature set does not grow during the feature search.

"Classes Only" The search space $\mathcal{P}$ begins with only unigram class features, and grows by dynamically contructing class n-gram questions as described earlier.

"Word and Classes" The search space $\mathcal{P}$ begins with all possible n-gram word features and unigram class features, and grows by adding class questions (as described earlier).

The results in Table 2 are achieved in the neighborhood of about 200 features. As can be seen in Figure 1, performance improves quickly as features are added and improves rather very slowly after the 60-th feature. The performance is fairly close for the various feature sets when a sufficient number of features are added. We also compared these results to a decision tree grown on the same 4 head-word events. The same

Experiment	Computer Manuals	Wall St. Journal
Words Only	82.2%	77.7%
Classes Only	84.5%	79.1%
Words and Classes	84.1%	81.6%

Table 2: Performance of ME Model on Test Events

Domain	Performance
Computer Manuals	79.5%
Wall St. Journal	77.7%

Table 3: Decision Tree Performance

mutual information bits were used for growing the decision trees. Table 3 gives the results on the same training and test data. The ME models are slightly better than the decision tree models.

For comparison, we obtained the PP-attachment performances of 3 treebanking experts on a set of 300 randomly selected test events from the WSJ corpus. In the first trial, they were given only the four head words to make the attachment decision, and in the next, they were given the headwords along with the sentence in which they occurred. Figure 3 shows an example of the head words test[4]. The results of the treebankers and the performance of the ME model on that same set are shown in Table 5. We also identified the set of 274 events on which treebankers, given the sentence, unanimously agreed. We defined this to be the truth set. We show in Table 6 the agreement on PP-attachment of the original WSJ treebank parses with this consensus set, the average performance of the 3 human experts with head words only, and the ME model. The WSJ treebank indicates the accuracy rate of our training data, the human performance indicates how much information is in the headwords, and the ME model is still a good 12 percentage points behind.

[4] the key is N,V,N,N,V, N,N,N,N,V,V,N,V,N,N,N,V,N,V

```
report milllion in charges
report milllion for quarter
reflecting settlement of contracts
carried all but one
were injuries among workers
had damage to building
be damage to some
uses variation of design
cited example of district
leads Pepsi in share
trails Pepsi in sales
risk conflict with U.S.
risk conflict over plan
oppose seating as delegate
save some of plants
introduced versions of cars
lowered bids in anticipation
oversees trading on Nasdaq
gained 1 to 19
```

Figure 3: Sample of 4 head words for PP-attachment

Selection Order	Feature
(1)	Preposition == "of"
(2)	Bit 2 of Head Noun == 0
(3)	Preposition is "to"
(4)	Bit 12 of Head Noun == 1
⋮	⋮
(9)	Head Noun == "million", Preposition == "in"
⋮	⋮
(30)	Preposition == "to", Bit 8 of Object == 1
⋮	⋮
(47)	Preposition == "in", Object == "months"

Table 4: Examples of Features Chosen for Wall St. Journal Data

Average Human(head words only)	88.2%
Average Human(with whole sentence)	93.2%
ME Model	78.0%

Table 5: Average Performance of Human & ME Model on 300 Events of WSJ Data

# Events in Consensus	% WSJ TB Performance	Human Performance	ME Model Performance
274	95.7%	92.5%	80.7%

Table 6: Human and ME model performance on consensus set for WSJ

We also obtained the performances of 3 non-experts on a set of 200 randomly selected test events from the Computer Manuals corpus. In this trial, the participants made attachment decisions given only the four head words. The results are shown in Table 7.

5. Conclusion

The Maximum Entropy model predicts prepositional phrase attachment 10 percentage points less accurately than a treebanker, but it performs comparably to a non-expert, assuming that only only the head words of the history are available in both cases. The biggest improvements to the ME model will come from better utilization of classes, and a larger history.

Currently, the use of the mutual information class bits gives us a few percentage points in performance, but the ME model should gain more from other word classing schemes which are better tuned to the PP-attachment problem. A scheme in which the word classes are built from the observed attachment preferences of words ought to outperform the mutual information clustering method, which uses only word bigram distributions[10].

Average Human	77.3%
ME Model	83.5%

Table 7: Average Performance of Human & ME Model on 200 Events of Computer Manuals Data

Secondly, the ME model does not use information contained in the rest of the sentence, although it is apparently useful in predicting the attachment, as evidenced by a 5% average gain in the treebankers' accuracy. Any implementation of this model using the rest of the sentence would require features on other words, and perhaps features on the sentence's parse tree structure, coupled with an efficient incremental search.

Such improvements should boost the performance of the model to that of treebankers. Already, the ME model outperforms a decision tree confronted with the same task. We hope to use Maximum Entropy to predict other linguistic phenomena that hinder the performance of most natural language parsers.

References

1. Jaynes, E. T., "Information Theory and Statistical Mechanics." *Phys. Rev.* **106**, pp. 620–630, 1957.
2. Kullback, S., *Information Theory in Statistics.* Wiley, New York, 1959.
3. Darroch, J.N. and Ratcliff, D., "Generalized Iterative Scaling for Log-Linear Models", *The Annals of Mathematical Statistics,* Vol. 43, pp 1470–1480, 1972.
4. Della Pietra,S., Della Pietra, V., Mercer, R. L., Roukos, S., "Adaptive Language Modeling Using Minimum Discriminant Estimation," *Proceedings of ICASSP-92,* pp. I-633-636, San Francisco, March 1992.
5. Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R., Nadas, A., and Roukos, S., "Maximum Entropy Methods and Their Applications to Maximum Likelihood Parameter Estimation of Conditional Exponential Models," *A forthcoming IBM technical report.*
6. Berger, A., Della Pietra, S.A., and Della Pietra, V.J.. Maximum Entropy Methods in Machine Translation. *manuscript in preparation*.
7. Black, E., Garside, R., and Leech, G., 1993. *Statistically-driven Computer Grammars of English: The IBM/Lancaster Approach.* Rodopi. Atlanta, Georgia.
8. Black, E., Jelinek, F., Lafferty, J., Magerman, D. M., Mercer, R., and Roukos, S., 1993. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In Proceedings of the Association for Computational Linguistics, 1993. Columbus, Ohio.
9. Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., 1984. *Classification and Regression Trees.* Wadsworth and Brooks. Pacific Grove, California.
10. Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. Class-based n-gram Models of Natural Language. In *Proceedings of the IBM Natural Language ITL,* March, 1990. Paris, France.
11. Hindle, D. and Rooth, M. 1990. Structural Ambiguity and Lexical Relations. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.
12. Magerman, D., 1994. Natural Language Parsing as Statistical Pattern Recognition. Ph. D. dissertation, Stanford University, California.