Decision Tree Models Applied to the Labeling of Text with Parts-of-Speech

Ezra Black *Fred Jelinek* *John Lafferty*

Robert Mercer *Salim Roukos*

IBM Thomas J. Watson Research Center

ABSTRACT

We describe work which uses decision trees to estimate marginal probabilities in a maximum entropy model for predicting the part-of-speech of a word given the context in which it appears. Two experiments are presented which exhibit improvements over the usual hidden Markov model approach.

1. Introduction

In this paper we describe work which uses decision trees to estimate probabilities of words appearing with various parts-of-speech, given the context in which the words appear. In principle, this approach affords the optimal solution to the problem of predicting the correct sequence of parts-of-speech. In practice, the method is limited by the lack of large, hand-labeled training corpora, as well as by the difficulty inherent in constructing a set of questions to be used in the decision procedure. Nevertheless, decision trees provide a powerful mechanism for tackling the problem of modeling long-distance dependencies.

The following sentence is typical of the difficulties facing a tagging program:

> *The new energy policy announced in December by the Prime Minister will guarantee sufficient oil supplies at one price only.*

The usual hidden Markov model, trained as described the last section of this paper, incorrectly labeled the verb *announced* as having the active rather than the passive aspect. If, however, a decision procedure is used to resolve the ambiguity, the context may be queried to determine the nature of the verb as well its agent. We can easily imagine, for example, that if the battery of available questions is rich enough to include such queries as "Is the previous noun inanimate?" and "Does the preposition *by* appear within three words of the word being tagged?" then such ambiguities may be probabilistically resolved. Thus it is evident that the success of the decision approach will rely in the questions as well as the manner in which they are asked.

In the experiments described in this paper, we constructed a complete set of binary questions to be asked of words, using a mutual information clustering procedure [2]. We then extracted a set of *events* from a 2-million word corpus of hand-labeled text. Using an algorithm similar to that described in [1], the set of contexts was divided into equivalence classes using a decision procedure which queried the binary questions, splitting the data based upon the principle of maximum mutual information between tags and questions. The resulting tree was then smoothed using the forward-backward algorithm [6] on a set of held-out events, and tested on a set of previously unseen sentences from the hand-labeled corpus.

The results showed a modest improvement over the usual hidden Markov model approach. We present explanations and examples of the results obtained and suggest ideas for obtaining further improvements.

2. Decision Trees

The problem at hand is to predict a tag for a given word in a sentence, taking into consideration the tags assigned to previous words, as well as the remaining words in the sentence. Thus, if we wish to predict tag t_n for word w_n in a sentence $S = w_1, w_2, \ldots, w_N$, then we must form an estimate of the probability

$$P(t_n \mid t_1, t_2, \ldots t_{n-1} \text{ and } w_1, w_2 \ldots, w_N) .$$

We will refer to a sequence $(t_1, \ldots, t_{n-1}; w_1, \ldots, w_N)$ as a *history*. A generic history is denoted as H, or as $\mathsf{H} = (\mathsf{H}_T, \mathsf{H}_W)$, when we wish to separate it into its tag and word components. The set of histories is denoted by $\mathcal{H}$, and a pair (t, H) is called an *event*.

A tag is chosen from a fixed tag vocabulary V_T, and words are chosen from a word vocabulary V_W. Given a training corpus $\mathcal{E}$ of events, the decision tree method proceeds by placing the observed histories into equivalence classes by asking binary questions about them. Thus, a tree is grown with each node labeled by a question $\mathtt{q} : \mathcal{H} \rightarrow \{\mathtt{True}, \mathtt{False}\}$. The entropy of tags at a

leaf L of the tree $\mathcal{T}$ is given by

$$H(T \mid L) = -\sum_{t \in T} P(t \mid L) \log P(t \mid L)$$

and the average entropy of tags in the tree is given by

$$\bar{H}_{\mathcal{T}}(T) = \sum_{L \in \mathcal{T}} P(L)\, H(T \mid L).$$

The method of growing trees that we have employed adopts a greedy algorithm, described in [1], to minimize the average entropy of tags.

Specifically, the tree is grown in the following manner. Each node n is associated with a subset $\mathcal{E}_n \subset \mathcal{E}$ of training events. For a given node n, we compute for each question q, the conditional entropy of tags at n, given by

$$\begin{aligned}\bar{H}(T \mid n, \mathsf{q}) = \\ P(\mathsf{q}(\mathsf{H}) = \texttt{True} \mid n)\, \bar{H}(T \mid n, \mathsf{q}(\mathsf{H}) = \texttt{True}) + \\ P(\mathsf{q}(\mathsf{H}) = \texttt{False} \mid n)\, \bar{H}(T \mid n, \mathsf{q}(\mathsf{H}) = \texttt{False}).\end{aligned}$$

The node n is then assigned the question q with the lowest conditional entropy. The reduction in entropy at node n resulting in asking question q is

$$\bar{H}(T \mid n) - \bar{H}(T \mid n, \mathsf{q}).$$

If this reduction is significant, as determined by evaluating the question on held-out data, then two descendent nodes of n are created, corresponding to the equivalence classes of events

$$\{\mathsf{E} = (t, \mathsf{H}) \mid \mathsf{E} \in \mathcal{E}_n,\ \mathsf{q}(\mathsf{H}) = \texttt{True}\}$$

and

$$\{\mathsf{E} = (t, \mathsf{H}) \mid \mathsf{E} \in \mathcal{E}_n,\ \mathsf{q}(\mathsf{H}) = \texttt{False}\}.$$

The algorithm continues to split nodes by choosing the questions which maximize the reduction in entropy, until either no further splits are possible, or until a maximum number of leaves is obtained.

3. Maximum Entropy Models

The above algorithm for growing trees has as its objective function the entropy of the joint distribution of tags and histories. More generally, if we suppose that tags and histories arise according to some distribution $\tilde{p}(t, \mathsf{H}_T, \mathsf{H}_W)$ in textual data, the coding theory point-of-view encourages us to try to construct a model for generating tags and histories according to a distribution $p(t, \mathsf{H}_T, \mathsf{H}_W)$ which minimizes the Kullback information

$$D(p \,||\, \tilde{p}) = \sum_{t, \mathsf{H}_T, \mathsf{H}_W} p(t, \mathsf{H}_T, \mathsf{H}_W) \log \frac{p(t, \mathsf{H}_T, \mathsf{H}_W)}{\tilde{p}(t, \mathsf{H}_T, \mathsf{H}_W)}.$$

Typically, one may be able to obtain estimates for certain marginals of p. In the case of tagging, we have estimates of the marginals $q(t, \mathsf{H}_T) = \sum_{\mathsf{H}_W} p(t, \mathsf{H}_T, \mathsf{H}_W)$ from the EM algorithm applied to labelled or partially labeled text. The marginals $r(t, \mathsf{H}_W) = \sum_{\mathsf{H}_T} p(t, \mathsf{H}_T, \mathsf{H}_W)$ might be estimated using decision trees applied to labelled text. To minimize $D(p \,||\, q)$ subject to knowing these marginals, introducing Lagrange multipliers α and β leads us to minimize the function

$$\begin{aligned}\sum_{t, \mathsf{H}_T, \mathsf{H}_W} p(t, \mathsf{H}_T, \mathsf{H}_W) \log \frac{p(t, \mathsf{H}_T, \mathsf{H}_W)}{\tilde{p}(t, \mathsf{H}_T, \mathsf{H}_W)} + \\ \sum_{t, \mathsf{H}_T} \alpha(t, \mathsf{H}_T) \left(\sum_{\mathsf{H}_W} p(t, \mathsf{H}_T, \mathsf{H}_W) - q(t, \mathsf{H}_T) \right) + \\ \sum_{t, \mathsf{H}_W} \beta(t, \mathsf{H}_W) \left(\sum_{\mathsf{H}_T} p(t, \mathsf{H}_T, \mathsf{H}_W) - r(t, \mathsf{H}_W) \right).\end{aligned}$$

Differentiating with respect to p and solving this equation, we find that the *maximum entropy* solution p takes the form

$$p(t, \mathsf{H}_T, \mathsf{H}_W) = \gamma f(t, \mathsf{H}_T) g(t, \mathsf{H}_W) \tilde{p}(t, \mathsf{H}_T, \mathsf{H}_W)$$

for some normalizing constant γ. In particular, in the case where we know no better than to take $\tilde{p}$ equal to the uniform distribution, we obtain the solution

$$p(t, \mathsf{H}_T, \mathsf{H}_W) = \frac{q(t, \mathsf{H}_T)\, r(t, \mathsf{H}_W)}{q(t)}$$

where the marginal $q(t)$ is assumed to satisfy

$$q(t) = \sum_{\mathsf{H}_T} q(t, \mathsf{H}_T) = \sum_{\mathsf{H}_W} r(t, \mathsf{H}_W).$$

Note that the usual HMM tagging model is given by

$$p(t_n, \mathsf{H}_T, \mathsf{H}_W) = \frac{P(t_n, t_{n-2}, t_{n-1})\ P(w_n, t_n)}{P(t_n)}$$

which has the form of a maximum entropy model, even though the marginals $P(w_n, t_n)$ and $P(t_n, t_{n-2}, t_{n-1})$ are modelled as bigram and trigram statistics, estimated according to the maximum likelihood criterion using the EM algorithm.

In principle, growing a decision tree to estimate the full density $p(t_n, \mathsf{H}_T, \mathsf{H}_W)$ will provide a model with smaller Kullback information. In practice, however, the quantity of training data is severely limited, and the statistics at the leaves will be unreliable. In the model described above we assume that we are able to construct more reliable estimates of the marginals separating the word

and tag components of the history, and we then combine these marginals according to the maximum entropy criterion. In the experiments that we performed, such models performed slightly better than those for which the full distribution $p(t_n, \mathsf{H}_T, \mathsf{H}_W)$ was modeled with a tree.

4. Constructing Questions

The method of mutual information clustering, described in [2], can be used to obtain a set of binary features to assign to words, which may in turn be employed as binary questions in growing decision trees. Mutual information clustering proceeds by beginning with a vocabulary V, and initially assigning each word to a distinct class. At each step, the average mutual information between adjacent classes in training text is computed using a bigram model, and two classes are chosen to be merged based upon the criterion of minimizing the loss in average mutual information that the merge affects. If this process is continued until only one class remains, the result is a binary tree, the leaves of which are labeled by the words in the original vocabulary. By labeling each branch by 0 or 1, we obtain a bit string assigned to each word.

Like all methods in statistical language modeling, this approach is limited by the problems of statistical significance imposed by the lack of sufficient training data. However, the method provides a powerful way of automatically extracting both semantic and syntactic features of large vocabularies. We refer to [2] for examples of the features which this procedure yields.

5. Smoothing the Leaf Distributions

After growing a decision tree according to the procedures outlined above, we obtain an equivalence class of histories together with an empirical distribution of tags at each leaf. Because the training data, which is in any case limited, is split exponentially in the process of growing the tree, many of the leaves are invariably associated with a small number of events. Consequently, the empirical distributions at such leaves may not be reliable, and it is desirable to smooth them against more reliable statistics.

One approach is to form the *smoothed distributions* $\widetilde{P}(\cdot \mid n)$ from the empirical distributions $P(\cdot \mid n)$ for a node n by setting

$$\widetilde{P}(t \mid n) = \lambda_n \, P(t \mid n) + (1 - \lambda_n) \, \widetilde{P}(t \mid \mathit{parent}(n))$$

where $\mathit{parent}(n)$ is the parent node of n (with the convention that $\mathit{parent}(\mathit{root}) = \mathit{root}$), and $0 \leq \lambda_n \leq 1$ can be thought of as the confidence placed in the empirical distribution at the node.

In order to optimize the coefficients λ_n, we seek to maximize the probability that the correct prediction is made for every event in a corpus $\mathcal{E}_H$ held-out from the training corpus used to grow the tree. That is, we attempt to maximize the objective function

$$\mathcal{O} = \prod_{(t, \mathsf{H}) \in \mathcal{E}_H} \widetilde{P}(t \mid L(\mathsf{H}))$$

as a function of the coefficients $\lambda = (\lambda_1, \lambda_2, \ldots)$ where $L(\mathsf{H})$ is the leaf of the history H. While finding the maximizing λ is generally an intractable problem, the EM algorithm can be adopted to estimate coefficients which locally maximize the above objective function. Since this is a straightforward application of the EM algorithm we will not present the details of the calculation here.

6. Experimental Results

In this section we report on two experiments in part-of-speech labeling using decision trees. In the first experiment, we created a model for tagging text using a portion of the Lancaster treebank. In the second experiment, we tagged a portion of the Brown corpus using a model derived from the University of Pennsylvania corpus of hand-corrected labeled text. In each case we compared the standard HMM model to a maximum entropy model of the form

$$\begin{aligned} &P(t_n \mid t_1, t_2, \ldots t_{n-1} \textit{ and } w_1, w_2 \ldots, w_N) = \\ &\quad = P(t_n \mid t_{n-2}, t_{n-1} \,;\, w_{n-2}, w_{n-1}, w_n, w_{n+1}, w_{n+2}) \\ &\quad = P(t_n \mid w_{n-2}, w_{n-1}, w_n, w_{n+1}, w_{n+2}) \times \\ &\qquad \times P(t_n \mid t_{n-2}, t_{n-1}) \, P(t_n)^{-1} \end{aligned}$$

where the parameters $P(t_n \mid t_{n-1}, t_{n-1})$ were obtained using the usual HMM method, and the parameters $P(t_n \mid w_{n-2}, w_{n-1}, w_n, w_{n+1}, w_{n+2})$ were obtained from a smoothed decision tree as described above. The trees were grown to have from 30,000 to 40,000 leaves.

The relevant data of the experiments is tabulated in Tables 2 and 3. The word and tag vocabularies were derived from the data, as opposed to being obtained from on-line dictionaries or other sources. In the case of the Lancaster treebank, however, the original set of approximately 350 tags, many of which were special tags for idioms, was compressed to a set of 163 tags. A rough categorization of these parts-of-speech appears in Table 1.

For training the model we had at our disposal approximately 1.9 million words of hand-labeled text. This corpus is approximately half AP newswire text and half English Hansard text, and was labeled by the team of Lancaster linguists. To construct our model, we divided the data into three sections, to be used for training, smooth-

29	Nouns
27	Verbs
20	Pronouns
17	Determiners
16	Adverbs
12	Punctuation
10	Conjunctions
8	Adjectives
4	Prepositions
20	Other

Table 1: Lancaster parts-of-speech

ing, and testing, consisting of 1,488,271 words, 392,732 words, and 51,384 words respectively.

We created an initial lexicon with the word-tag pairs that appear in the training, smoothing, and test portions of this data. We then filled out this lexicon using a statistical procedure which combines information from word spellings together with information derived from word bigram statistics in English text. This technique can be used both to discover parts-of-speech for words which do not occur in the hand-labeled text, as well as to discover additional parts-of-speech for those that do. In both experiments multiword expressions, such as "nineteenth-century" and "stream-of-consciousness," which were assigned a single tag in the hand-labelled text, were broken up into single words in the training text, with each word receiving no tag.

The parameters of the HMM model were estimated from the training section of the hand-labeled text, without any use of the forward-backward algorithm. Subsequently, we used the smoothing section of the data to construct an interpolated model as described by Merialdo [4, 6].

We evaluated the performance of the interpolated hidden Markov model by tagging the 2000 sentences which make up the testing portion of the data. We then compared the resultant tags with those produced by the Lancaster team, and found the error rate to be 3.03%.

We then grew and smoothed a decision tree using the same division of training and smoothing data, and combined the resulting marginals for predicting tags from the word context with the marginals for predicting tags from the tag context derived from the HMM model. The resulting error rate was 2.61%, a 14% reduction from the HMM model figure.

Tag vocabulary size:	163 tags
Word vocabulary size:	41471 words
Training data:	1,488,271 words
Held-out data:	392,732 words
Test data:	51,384 words (2000 sentences)
Source of data:	Hansards AP newswire
Dictionary:	no unknown words
Multiword expressions:	broken up
HMM errors:	1558 (3.03%)
Decision tree errors:	1341 (2.61%)
Error reduction:	13.9%

Table 2: Lancaster Treebank Experiment

In the case of the experiment with the UPenn corpus, the word vocabulary and dictionary were derived from the training and smoothing data only, and the dictionary was not statistically filled out. Thus, there were unknown words in the test data. The tag set used in the second experiment was comprised of the 48 tags chosen by the UPenn project. For training the model we had at our disposal approximately 4.4 million words of hand-labeled text, using approximately half the Brown corpus, with the remainder coming from the Wall Street Journal texts labelled by the UPenn team. For testing the model we used the remaining half of the Brown corpus, which was not used for any other purpose. To construct our model, we divided the data into a training section of 4,113,858 words, and a smoothing section of 292,731 words. The error rate on 8,000 sentences from the Brown corpus test set was found to be 4.57%. The corresponding error rate for the model using a decision tree grown only on the Brown corpus portion of the training data was 4.37%, representing only a 4.31% reduction in the error rate.

7. Conclusions

In two experiments we have seen how decision trees provide modest improvements over HMM's for the problem of labeling unrestricted text with parts-of-speech. In examining the errors made by the models which incorporate the decision tree marginals, we find that the errors may be attributed to two primary problems: bad ques-

Tag vocabulary size:	48 tags
Word vocabulary size:	86456 words
Training data:	4,113,858 words
Held-out data:	292,731 words
Test data:	212,064 words (8000 sentences)
Source of data:	Brown corpus Wall Street Journal
Dictionary:	unknown test words
Multiword expressions:	broken up
HMM errors:	9683 (4.57%)
Decision tree errors:	9265 (4.37%)
Error reduction:	4.31%

Table 3: UPenn Brown Corpus Experiment

tions and insufficient training data. Consider the word *lack*, for example, which may be either a noun or a verb. The mutual information clustering procedure tends to classify such words as either nouns or verbs, rather than as words which may be both. In the case of *lack* as it appeared in the Lancaster data, the binary features emphasized the nominal apects of the word, relating it to such words as *scarcity, number, amount* and *portion.* This resulted in errors when it occurred as a verb in the test data.

Clearly an improvement in the binary questions asked of the histories is called for. In a preliminary set of experiments we augmented the automatically-derived questions with a small set of hand-constructed questions which were intended to resolve the ambiguity of the label for verbs which may have either the active or passive aspect. The resulting decision trees, however, did not significantly improve the error rate on this particular problem, which represents inherently long-distance linguistic phenomena. Nevertheless, it appears that the basic approach can be made to prosper through a combination of automatic and linguistic efforts.

References

1. L. Bahl, P. Brown, P. deSouza, and R. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37, pp. 1001-1008, 1989.
2. P. Brown, V. Della Pietra, P. deSouza, and R. Mercer. Class-based n-gram models of natural language. *Proceedings of the IBM Natural Language ITL*, pp. 283-298, Paris, France, 1990.
3. K. Church. A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.
4. B. Merialdo. Tagging text with a probabilistic model. *IBM Research Report, RC 15972*, 1990.
5. M. Meteer, R. Schwartz, and R. Weischedel. Studies in part of speech labelling. In Proceedings of the February 1991 DARPA Speech and Natural Language Workshop. Asilomar, California.
6. S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35, Number 3, pp. 400-401, 1987.