

Peter A. Heeman

France Télécom CNET
Technopole Anticipa - 2 Avenue Pierre Marzin
22301 Lannion Cedex, France.
heeman@lannion.cnet.fr

James F. Allen

Department of Computer Science
University of Rochester
Rochester NY 14627, USA
james@cs.rochester.edu

ABSTRACT

Language models for speech recognition tend to concentrate solely on recognizing the words that were spoken. In this paper, we redefine the speech recognition problem so that its goal is to find both the best sequence of words *and* their syntactic role (part-of-speech) in the utterance. This is a necessary first step towards tightening the interaction between speech recognition and natural language understanding.

1 INTRODUCTION

For recognizing spontaneous speech, the acoustic signal is too weak to narrow down the number of word candidates. Hence, speech recognizers employ a language model that prunes out acoustic alternatives by taking into account the previous words that were recognized. In doing this, the speech recognition problem is viewed as finding the most likely word sequence \hat{W} given the acoustic signal [11].

$$\begin{aligned} \hat{W} &= \arg \max_W \Pr(W|A) \\ &= \arg \max_W \frac{\Pr(A|W) \Pr(W)}{\Pr(A)} \\ &= \arg \max_W \Pr(A|W) \Pr(W) \end{aligned}$$

The last line involves two probabilities that need to be estimated—the first due to the acoustic model $\Pr(A|W)$ and the second due to the language model $\Pr(W)$. The probability due to the language model can be expressed as the following, where we rewrite the sequence W explicitly as the sequence of N words $W_{1,N}$.

$$\Pr(W_{1,N}) = \prod_{i=1}^N \Pr(W_i|W_{1,i-1})$$

To estimate the probability distribution, a training corpus is typically used from which the probabilities can be estimated by relative frequencies. Due to sparseness of data, one must define equivalence classes amongst the contexts $W_{1,i-1}$, which can be done by limiting the context to an n -gram language model [11] and also by grouping words into words classes [4].

Several attempts have been made to incorporate shallow syntactic information to give better equivalence classes, where the shallow syntactic information is expressed as part-of-speech (POS) tags (e.g. [11], [13]). A POS tag

indicates the syntactic role that a particular word is playing in the utterance, e.g. whether it is a noun or a verb, etc. The approach is to use the POS tags of the prior few words to define the equivalence classes. This is done by summing over all POS possibilities as shown below.

$$\begin{aligned} \Pr(W_i|W_{1,i-1}) &= \sum_{P_{1,i}} \Pr(W_i|P_{1,i}W_{1,i-1}) \Pr(P_{1,i}|W_{1,i-1}) \\ &= \sum_{P_{1,i}} \Pr(W_i|P_{1,i}W_{1,i-1}) \Pr(P_i|P_{1,i-1}W_{1,i-1}) \Pr(P_{1,i-1}|W_{1,i-1}) \end{aligned}$$

Furthermore, the following two assumptions are made to simplify the context.

$$\begin{aligned} \Pr(W_i|P_{1,i}W_{1,i-1}) &\approx \Pr(W_i|P_i) \\ \Pr(P_i|P_{1,i-1}W_{1,i-1}) &\approx \Pr(P_i|P_{1,i-1}) \end{aligned}$$

However, this approach does not lead to an improvement in the performance of the speech recognizer. For instance, Srinivas [15] reports that such a model results in a 24.5% increase in perplexity over a word-based model on the Wall Street Journal, and Niesler and Woodland [13] report an 11.3% increase (but a 22-fold decrease in the number of parameters of such a model). Only by interpolating in a word-based model is an improvement seen [11].

A more major problem with the above approach is that in a spoken dialogue system, speech recognition is only the first step in understanding a speaker's contribution. One also needs to determine the syntactic structure of the words involved, its semantic meaning, and the speaker's intention in making the utterance. This information is needed to help the speech recognizer constrain the alternative hypotheses. Hence, we need a tighter coupling between speech recognition and the rest of the interpretation process.

2 REDEFINING THE PROBLEM

As a starting point, we re-examine the approach of using POS tags in the speech recognition process. Rather than view POS tags as intermediate objects solely to find the best word assignment, we redefine the goal of the speech recognition process so that it finds the best word sequence *and* the best POS interpretation given the acoustic signal.

$$\begin{aligned} \hat{W}\hat{P} &= \arg \max_{WP} \Pr(WP|A) \\ &= \arg \max_{WP} \Pr(A|WP) \Pr(WP) \end{aligned}$$

The first term $\Pr(A|WP)$ is the acoustic model, which traditionally excludes the category assignment. The second term $\Pr(WP)$ is the POS-based language model. Just as

*This research work was completed while the first author was at the University of Rochester. The authors would like to thank Geraldine Damnati, Kyung-ho Loken-Kim, Tsuyoshi Morimoto, Eric Ringger and Ramesh Sarukkai. This material is based upon work supported by the NSF under grant IRI-9623665 and by ONR under grant N00014-95-1-1088.

before, we rewrite the probability of $\Pr(WP)$ as a product of probabilities of the word and POS tag given the previous context.

$$\begin{aligned} \Pr(W_{1,N} P_{1,N}) &= \prod_{i=1,j} \Pr(W_i P_i | W_{1,i-1} P_{1,i-1}) \\ &= \prod_{i=1,j} \Pr(W_i | W_{1,i-1} P_{1,i}) \Pr(P_i | W_{1,i-1} P_{1,i-1}) \end{aligned}$$

The final probability distributions are similar to those used for POS tagging of written text [5, 6, 7]. However, these approaches simplify the probability distributions as is done by previous attempts to use POS tags in speech recognition language models.¹ As we will show in Section 4.1, such simplifications lead to poorer language models.

3 ESTIMATING THE PROBABILITIES

The probability distributions that we now need to estimate are more complicated than the traditional ones. Our approach is to use the decision tree learning algorithm [1, 2, 3], which uses information theoretic measures to construct equivalence classes of the context in order to cope with sparseness of data. The decision tree algorithm starts with all of the training data in a single leaf node. For each leaf node, it looks for the question to ask of the context such that splitting the node into two leaf nodes results in the biggest decrease in *impurity*, where the impurity measures how well each leaf predicts the events in the node. Heldout data is used to decide when to stop growing the tree: a split is rejected if the split does not result in a decrease in impurity with respect to the heldout data. After the tree is grown, the heldout dataset is used to smooth the probabilities of each node with its parent [1].

3.1 Word and POS Classification Trees

To allow the decision tree to ask about the words and POS tags in the context, we cluster the words and POS tags using the algorithm of Brown *et al.* [4] into a binary classification tree. The algorithm starts with each word (or POS tag) in a separate class, and successively merges classes that result in the smallest loss in mutual information in terms of the co-occurrences of these classes. By keeping track of the order that classes were merged, we can construct a hierarchical classification of the words. Figure 1 shows a classification tree that we grew for the POS tags. The binary classification tree gives an implicit binary encoding for each word and POS tag, which we show after each POS tag in the figure. The decision tree algorithm can then ask questions about the binary encoding of the words, such as ‘is the third bit of the POS tag encoding equal to one?’, and hence can ask about which partition a word is in.

Unlike other work that uses classification trees as the basis for the questions used by a decision tree (e.g. [2]), we treat the word identities as a further refinement of the POS tags. This approach has the advantage of avoiding unnecessary data fragmentation, since the POS tags and word identities will not be viewed as separate sources of information. We grow the classification tree by starting with a unique class for each word and each POS tag that

¹A notable exception is the work of Black *et al.* [2], who use a decision tree to learn the probability distributions for POS tagging.

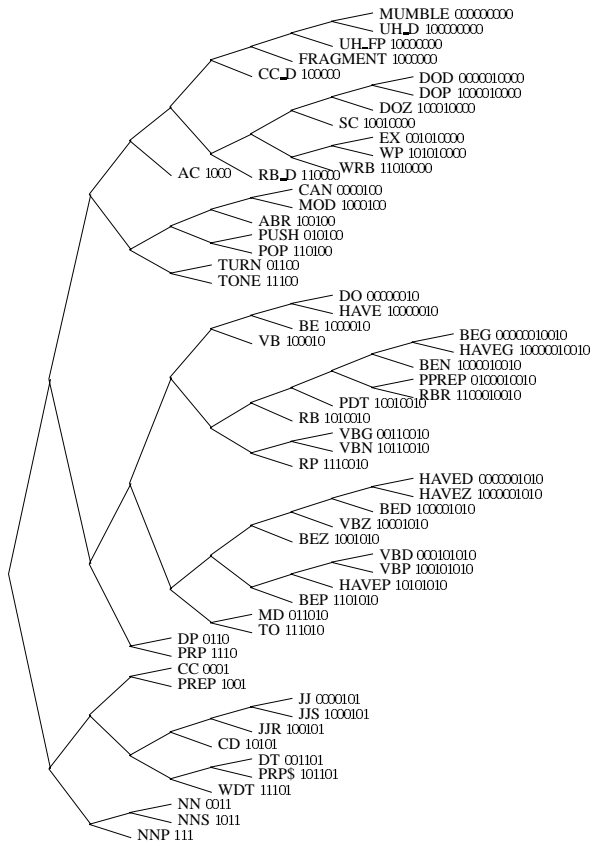


Figure 1: POS Classification Tree

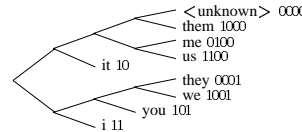


Figure 2: A Word Classification Tree

it takes on. When we merge classes to form the hierarchy, we only allow merges if all of the words in both classes have the same POS tag. The result is a word classification tree for each POS tag. This approach to growing the word trees simplifies the task, since we can take advantage of the hand-coded linguistic knowledge (as represented by the POS tags). Furthermore, we can better deal with words that can take on multiple senses, such as the word ‘loads’, which can be a plural noun (**NNS**) or a present tense third-person verb (**PRP**).²

In Figure 2, we give the classification tree for the personal pronouns (**PRP**). It is interesting to note that the clustering algorithm distinguished between the subjective pronouns ‘I’, ‘we’, and ‘they’, and the objective pronouns ‘me’, ‘us’, and ‘them’. The pronouns ‘you’ and ‘it’ can take either case, and the algorithm partitioned them according to their most common usage in the training corpus. Although distinct POS tags could have been added to distinguish between these two cases, it seems that the clustering algorithm can make up for some of the shortcomings

²Words-POS combinations that occur only once in the training corpus are grouped together in the class <unknown>, which is unique for each POS tag.

of the tagset.³

3.2 Composite Questions

In the previous section, we discussed the elementary questions that can be asked of the words and POS tags in the context. However, there might be a relevant partitioning of the data that can not be expressed in that form. For instance, a good partitioning of a node might involve asking whether questions q_1 and q_2 are both true. Using elementary questions, the decision tree would need to first ask question q_1 and then ask q_2 in the true subnode created by q_1 . This means that the false case has been split into two separate nodes, which could cause unnecessary data fragmentation.

Unnecessary data fragmentation can be avoided by allowing composite questions. Bahl *et al.* [1] introduced a simple but effective approach for constructing composite questions. Rather than allowing any boolean combination of elementary questions, they restrict the typology of the combinations to *pylons*, which have the following form (*true* maps all data into the true subset).

$$\begin{aligned} \text{pylon} &\Rightarrow \text{true} \\ \text{pylon} &\Rightarrow (\text{pylon} \wedge \text{elementary}) \\ \text{pylon} &\Rightarrow (\text{pylon} \vee \text{elementary}) \end{aligned}$$

The effect of any binary question is to divide the data into true and false subsets. The advantage of pylons is that each successive elementary question has the effect of swapping data from the true subnode into the false or vice versa. Hence, one can compute the change in node impurity that results from each successive elementary question that is added. This allows one to use a greedy algorithm to build the pylon by successively choosing the elementary question that results in the largest decrease in node impurity.

We actually employ a beam search and explore the best 10 alternatives at each level of the pylon. Again we make use of the heldout data to help pick the best pylon, but we must be careful not to make too much use of it for otherwise it will become as biased as the training data. If the last question added to a candidate pylon results in an increase in node impurity with respect to the heldout data, we remove that question and stop growing that alternative. When there are no further candidates that can be grown, we choose the winning pylon as the one with the best decrease in node impurity with respect to the training data. The effect of using composite questions is explored in Section 4.3.

4 RESULTS

To demonstrate our model, we have tested it on the Trains corpus [9], a collection of human-human task-oriented spoken dialogues consisting of 6 and half hours worth of speech, 34 different speakers, 58,000 words of transcribed speech, with a vocabulary size of 860 words. To make the best use of the limited amount of data, we use a 6-fold cross validation procedure, in which we use each sixth of the corpus for testing data, and the rest for training data.

³The words included in the <unknown> class are the reflexive pronouns ‘themselves’, and ‘itself’, which each occurred once in the training corpus.

A way to measure a language model is to compute the *perplexity* it assigns to a test corpus, which is an estimate of how well the language model is able to predict the next word. The perplexity of a test set of N words $w_{1,N}$ is calculated as follows,

$$2^{-\frac{1}{N} \sum_{i=1}^N \log_2 \hat{P}_r(w_i | w_{1,i-1})}$$

where \hat{P}_r is the probability distribution supplied by the language model. Full details of how we compute the word-based perplexity are given in [8]. We also measure the error rate in assigning the POS tags. Here, as in measuring the perplexity, we run the language model on the hand-transcribed word annotations.

4.1 Effect of Richer Context

Table 1 gives the perplexity and POS tagging error rate (expressed as a percent). To show the effect of the richer modeling of the context, we vary the amount of context given to the decision tree. As shown by the perplexity results, the context used for traditional POS-based language models (second column) is very impoverished. As we remove the simplifications to the context, we see the perplexity and POS tagging rates improve. By using both the previous words and previous POS tags as the context, we achieve a 43% reduction in perplexity and a 5.4% reduction in the POS error rate.

Context for W_i	P_i	$P_{i-3,i}$	$P_{i-3,i}W_{i-3,i-1}$	$P_{i-3,i}W_{i-3,i-1}$
Content for P_i	$P_{i-3,i-1}$	$P_{i-3,i-1}$	$P_{i-3,i-1}$	$P_{i-3,i-1}W_{i-3,i-1}$
POS Error Rate	3.13	3.10	3.03	2.97
Perplexity	42.32	32.11	29.49	24.17

Table 1: Using Richer Contexts

4.2 Constraining the Decision Tree

As we mentioned earlier, the word identity information W_{i-j} is viewed as further refining the POS tag of the word P_{i-j} . Hence, questions about the word encoding are only allowed if the POS tag is uniquely defined. Furthermore, for both POS and word questions, we restrict the algorithm so that it only asks about more specific bits of the POS tag and word encodings only if it has already uniquely identified the less specific bits. In Table 2, we contrast the effectiveness of adding further constraints. The second column gives the results of adding no further constraints, the third column only allows questions about a POS tag P_{i-j-1} only if P_{i-j} is uniquely determined, and the fourth column adds the constraint that the word W_{i-j} must also be uniquely identified before questions are allowed of P_{i-j-1} .

From the table, we see that it is worthwhile to force the decision tree to fully explore a POS tag for a word in the context before asking about previous words. Hence, we see that the decision tree algorithm needs help in learning that it is better to fully explore the POS tags. However,

	None	POS	Full
POS Error Rate	3.19	2.97	3.00
Perplexity	25.64	24.17	24.39

Table 2: Adding Additional Constraints

we see that adding the further constraint that the word identity should also be fully explored results in a decrease in performance of the model. Hence, we see that it is not worthwhile for the decision tree to fully explore the word information (which is the basis of class-based approaches to language modeling), and it is able to learn this on its own.

4.3 Effect of Composites

The next area we explore is the benefit of composite questions in estimating the probability distributions. The second column of Table 3 gives the results if composite questions are not employed, the third column gives the results if composite questions are employed, and the fourth gives the results if we employ a beam search in finding the best pylon (with up to 10 alternatives). From the results, we see that the use of pylons reduces the word perplexity rate by 4.7%, and the POS error rate by 2.3%. Furthermore, we see that using a beam search, rather than an entirely greedy algorithm accounts for some of the improvement.

	Not Used	Single	10
POS Error Rate	3.04	3.04	2.97
Perplexity	25.36	24.36	24.17

Table 3: Effect of Composite Questions

4.4 Effect of Larger Context

In Table 4, we look at the effect of the size of the context, and compare the results to a word-based backoff language model [12] built using the CMU toolkit [14]. For a bigram model, it has a perplexity of 29.3, in comparison to our word perplexity of 27.4. For a trigram model, the word-based model has a perplexity of 26.1, in comparison to our perplexity of 24.2. Hence we see that our POS-based model results in a 7.2% improvement in perplexity.

	Bigram	Trigram	4-gram
POS Error Rate	3.19	2.97	2.97
Perplexity	27.37	24.26	24.17
Word-based Model	29.30	26.13	

Table 4: Using Larger Contexts

5 CONCLUSION

In this paper, we presented a new way of incorporating POS information into a language model. Rather than treating POS tags as intermediate objects solely for recognizing the words, we redefine the speech recognition problem so that its goal is to find the best word sequence *and* their best POS assignment. This approach allows us to use the POS tags as part of the context for estimating the probability distributions. In fact, we view the word identities in the context as a refinement of the POS tags rather than viewing the POS tags and word identities as two separate sources of information. To deal with this rich context, we make use of decision trees, which can use information theoretic measures to automatically determine how to partition the contexts into equivalence classes. We find that this model results in a 7.2% reduction in perplexity over a

trigram word-based model for the Trains corpus of spontaneous speech. Currently, we are exploring the effect of this model in reducing the word error rate.

Incorporating shallow syntactic information into the speech recognition process is just the first step. In other work [8, 10], this syntactic information, as well as the techniques introduced in this paper, are used to help model the occurrence of dysfluencies and intonational phrasing in a speech recognition language model. Our use of decision trees to estimate the probability distributions proves effective in dealing with the richer context provided by modeling these spontaneous speech events. Modeling these events improves the perplexity to 22.5, a 14% improvement over the word-based trigram backoff model, and reduces the POS error rate by 9%.

6 REFERENCES

- [1] L. R. Bahl, P. F. Brown, P. V. deSouza, and R. L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7), 1989.
- [2] E. Black, F. Jelinek, J. Lafferty, R. Mercer, and S. Roukos. Decision tree models applied to the labeling of text with parts-of-speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 117–121, 1992.
- [3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, CA, 1984.
- [4] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n -gram models of natural language. *Computational Linguistics*, 18(4), 1992.
- [5] E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz. Equations for part-of-speech tagging. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '93)*, 1993.
- [6] K. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 136–143, 1988.
- [7] S. J. DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39, 1988.
- [8] P. A. Heeman. Speech repairs, intonational boundaries and discourse markers: Modeling speakers' utterances in spoken dialog. Doctoral dissertation. 1997.
- [9] P. A. Heeman and J. F. Allen. The Trains spoken dialog corpus. CD-ROM, Linguistics Data Consortium, 1995.
- [10] P. A. Heeman and J. F. Allen. Intonational boundaries, speech repairs, and discourse markers: Modeling spoken dialog. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.
- [11] F. Jelinek. Self-organized language modeling for speech recognition. Technical report, IBM T.J. Watson Research Center, Continuous Speech Recognition Group, 1985.
- [12] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 400–401, March 1987.
- [13] T. R. Niesler and P. C. Woodland. A variable-length category-based n -gram language model. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, 1996.
- [14] R. Rosenfeld. The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, 1995.
- [15] B. Srinivas. "Almost parsing" techniques for language modeling. In *Proceedings of the 4rd International Conference on Spoken Language Processing*, 1996.