AN INTEGRATED GRAMMAR/BIGRAM LANGUAGE MODEL USING PATH SCORES

Harvey Lloyd-Thomas, Jerry H.Wright*† and Gareth J.F.Jones‡*

* Ensigma Limited, Turing House, Station Road, Chepstow, NP6 5PB, U.K.
† Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, U.K.
‡ Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, U.K.

ABSTRACT

This paper describes a language model in which context-free grammar rules are integrated into an *n*-gram framework, complementing it instead of attempting to replace it. This releases the grammar from the aim of parsing sentences overall (which is often undesirable as well as unrealistic), enabling it to be employed selectively in modelling phrases that are identifiable within a flow of speech. Algorithms for model training, and for sentence scoring and interpretation are described. All are based on the principle of summing over paths that span the sentence, but implementation is node-based for efficiency. Perplexity results for this system (using a hierarchy of grammars from empty to full-coverage) are compared with those for *n*-gram models, and the system is used for re-scoring *N*-best sentence lists for a speaker-independent recogniser.

1. INTRODUCTION

Context-free grammars and *n*-grams are often regarded as alternative kinds of language model, but they have qualities that can complement each other. This paper describes an integrated model, first in a training procedure for symbol bigram and grammar-rule probabilities, and then in scoring and interpretation procedures. By placing the emphasis upon phrasal syntactic structure, and removing the requirement that a sentence parse overall, the aim is to enable the grammar to enhance the recognition of meaningful phrases within sentences that may be ill-formed.

In previous work [1] we have reported results for a hybrid recognition system in which grammar and bigram models operate in parallel. Sentences are implicitly partitioned into two classes, with consequent problems for interpreting and comparing scores across classes. We have also experimented with "extended" bigrams and trigrams [2,3], which emphasise the importance of structures wider in scope than conventional trigrams. Other researchers have reached similar conclusions [e.g. 4,5], and previous work combining *n*-gram and CFG models [6,7] has proved the significance of this aim.

In general, the chart of syntactic structures detected within a sentence will involve subtree sharing and local ambiguity packing, figure 1. A score for such a structure could be inherited from that of a top-level path that spans the sentence, connecting syntactic nodes. If bigrams are extended to cover nonterminal (as well as terminal or pre-terminal) symbols, the score is essentially given by

$$\prod P(\text{symbol} \mid \text{previous symbol}) P(\text{derivation from symbol})$$

Three problems with this are the following:

- the top-level path may not be unique (there are two such paths shown as dashed lines in figure 1),
- the score is remote from the words,
- the score is very sensitive to changes to the grammar.

Conceptually however, spanning paths can be traced through all levels of the structure, a score assigned to each by the same principle, and the sentence score inferred as the total over all paths. The dotted lines in figure 1 show the additional bigram links needed. This alleviates the three problems mentioned, in that the total score is unique (and the language model is normalised), it includes scores from paths that are close to the words, and many of the paths remain substantially unchanged through changes to the grammar.

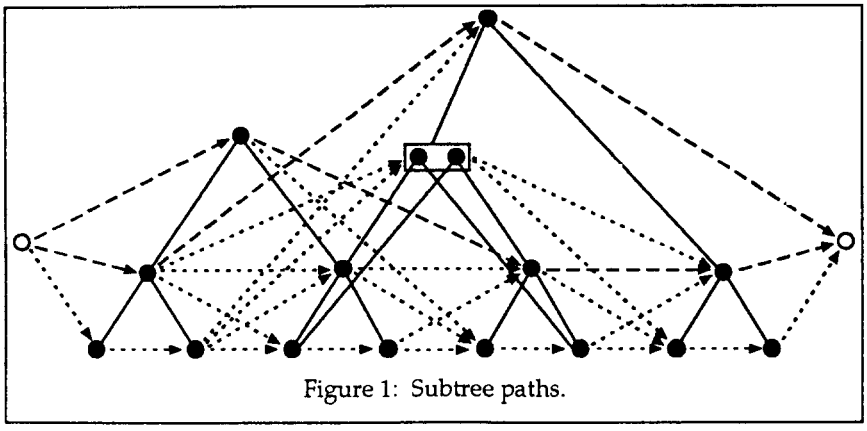

Figure 1: Subtree paths.

In practice there can be billions of paths

0-7803-2431-5/95 $4.00 © 1995 IEEE

through a long sentence, so node-based procedures must be used for training, scoring and interpretation.

2. TRAINING PROCEDURE

We assume that there is a given set of grammar rules established in advance, the grammar can consist of a disjoint set of subgrammars. Each sentence in the training corpus is parsed (as far as possible): we use a generalized LR algorithm for this purpose. Then a two-pass algorithm finds the number of spanning paths that pass through each node and along each bigram link; full details are given in [8]. By accumulating over the corpus the proportion for each sentence of total paths that use each grammar rule (taking account of local ambiguity) and bigram, elements of the model are credited in proportion to their usage in modelling the training data. A special procedure handles null rules. Grammar-rule and symbol bigram probabilities are then found by normalising and smoothing We are currently experimenting with different smoothing procedures.

A single pass over the training corpus is sufficient for this procedure, although a probability re-estimation version could be devised.

3. SENTENCE SCORING PROCEDURE

Let $\text{span}(X)=(j,k)$ denote the part of the sentence spanned by node X, where $1 \le j \le k \le L$ for sentence length L. Let \$ denote an end-of-sentence marker, so the actual sentence string is $\$w_1 w_2 \cdots w_L \$$. The following (similar to the HMM forward algorithm) finds the overall score.

(1) For each node Y such that $\text{span}(Y)=(1,m)$ for some m,

$$\alpha(Y,m) = P(Y|\$)\,P(Y \stackrel{*}{\Rightarrow} w_1 \cdots w_m)$$

(2) For all k from 2 to L, and for each node Y such that $\text{span}(Y)=(j,k)$ for some $j>1$, if $X_1,\cdots,X_n$ are all the nodes such that $\text{span}(X_i)=(m_i,j-1)$ for some m_i then

$$\alpha(Y,k) = \left[\sum_{i=1}^{n} \alpha(X_i, j-1)\,P(Y|X_i)\right] P(Y \stackrel{*}{\Rightarrow} w_j \cdots w_k)$$

(3) If $X_1,\cdots,X_n$ are all the nodes such that $\text{span}(X_i)=(m_i,L)$ for some m_i then

$$P(\$w_1 w_2 \cdots w_L \$) = \sum_{i=1}^{n} \alpha(X_i, L)\,P(\$|X_i)$$

Derivation probabilities of the form $P(X \stackrel{*}{\Rightarrow} x)$ include the sum over all local ambiguities within the subtree(s) dominated by X, and are inferred from the output of the substring parser. If X is a terminal node then this probability can be set to 1 (for perplexity calculations) or to the word acoustic likelihood (for recognition). $P(Y|X)$ is the bigram probability.

4. SENTENCE INTERPRETATION

We can now reconsider the top-level paths (the dashed lines in figure 1). Each provides an interpretation of the sentence as a sequence of phrases. We define a "trail" as a path for which no substring is reducible to a higher-level node, and in general a sentence can have many trails. To be consistent with sentence scoring, we assign a trail score as the sum of scores for paths bounded above by the trail.

The procedure for this is more complex than that for the sentence score. There are two conditions to satisfy: first, a trail must not pass through all the daughter nodes of any other node, and second, trail scoring must skip nodes not dominated by the trail nodes. Both of these conditions can be handled by appropriate book-keeping. As the trail highlighted in bold in figure 2 demonstrates, it is possible for all the nodes along a trail to be reduced, which can make a trail difficult to find efficiently, but the latest version of this procedure finds and scores the trails with only a modest additional overhead. We have space here for only a sketch of the procedure.

Define a node as "L-reduced", "M-reduced" or "R-reduced" if it is reduced as (respectively) a left-most, intermediate, or right-most subtree to a higher structure. With each reduce action we associate a unique identifying number, and for each node Y we construct lists L_Y, M_Y, R_Y of reduction identifiers for which Y is L-, M-, and R-reduced respectively. A node X is "marked" if it is either unreduced, or shared, or adjacent to another marked node Y (with $\text{span}(Y)=(j,k)$) in that either

- $\text{span}(X)=(m,j-1)$ for some m and X is not R-reduced,
- $\text{span}(X)=(k+1,m)$ for some m and X is not L-reduced.

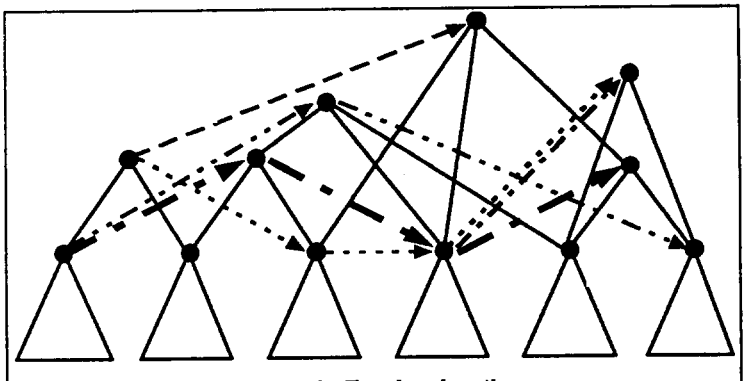

Figure 2: Top-level trails.

Language model	*Number of rules*	*Number of nonterminals*	*Test corpus perplexity*	*Trigram corpus perplexity*
Unigram	–	–	116.5	103.9
Bigram	–	–	17.1	13.4
Trigram	–	–	12.9	9.1
Extended bigram	–	–	12.3	19.0
Extended trigram	–	–	8.9	18.3
Grammar, full	228	288	12.2	21.7
Grammar, inter1	181	267	12.8	22.0
Grammar, inter2	174	260	12.8	22.0
Grammar, inter3	72	205	12.5	21.0
Grammar, empty	0	1	34.6	21.5

Table 1: ARM corpus perplexities

Adjacent marking is propagated using a simple two-pass algorithm, and trails pass only through marked nodes.

For each trail t we maintain a set S_t of identifiers. When the trail meets a marked node Y, we add all new reduction identifiers for which Y is L-reduced, and delete all earlier identifiers for which Y is not M-reduced, by updating S_t to $(S_t \cap M_Y) \cup L_Y$. If a reduction identifier is present in the set when the right-most node is reached then the trail must have traversed all the daughter nodes and is discontinued. This is indicated by $S_t \cap R_Y \neq \emptyset$.

With each node we also associate a list of identifiers for the top-most reductions that dominate it. Using this it is easy to check whether or not a node is within the same subtree as a trail node. The trail score is the sum of scores for all paths through nodes dominated by the trail nodes, and is found by an approach similar to that for the sentence score. To save time, node marking, trail control and scoring all occur within a single procedure.

Because each trail consists of a unique sequence of nodes it would be possible to apply either higher-order *n*-grams or extended *n*-grams [2,3] to score this sequence. This may improve the capability of the system to find the best interpretation and is the subject of current work.

5. RESULTS

Results have been obtained using a corpus of Airborne Reconnaissance Mission (ARM) reports [9]. These have a vocabulary of 511 words, and each report consists of a series of sentences of standard types. There is a full grammar for these reports, which we have adapted into context-free form. This allows us to study the effect of using a hierarchy of grammars, from empty to full coverage. Previous work using this corpus [2,3] has shown that significant reductions in test corpus perplexity are achieved using extended bigrams and trigrams, compared with their conventional counterparts.

5.1 Corpus perplexity

Table 1 contains perplexity figures for a test corpus of ARM sentences, scored using a hierarchy of *n*-gram models from basic unigram to extended bigram and trigram. The value 12.2 obtained using the full-coverage grammar compares well with the results obtained using *n*-grams. The low perplexity persists when rules are progressively removed from the grammar, (the model is re-trained for each grammar). The empty grammar essentially acts as a bigram model, but with different smoothing than for the word bigram model, which accounts for the higher perplexity. The figures in the bottom half of the table are dependent on the smoothing procedure, and there is also some pruning of paths in the current implementation.

Test corpus perplexity is higher than that for the extended trigram model, even with the full grammar. This is partly attributable to the fact that paths that are close to the words, and therefore involve a lot of bigrams, make a major contribution to the sentence score. This is not necessarily a disadvantage, and it would be easy to give greater weight to higher-level paths if desired.

For comparison, corresponding figures are shown for a corpus generated at random using the smoothed trigram

model, and as expected these are higher.

5.2 Re-scoring of *N*-best lists

N-best lists are inferred from the word lattices generated by the speaker-independent continuous ARM recogniser (from acoustic scores with no initial language model). Each is a report consisting of several consecutive sentences, with a total length of around 50 words. For this reason, the true report is often not in the *N*-best list, even for $N = 100$. We therefore score each candidate report and correlate the negative log score with the minimum distance (number of insertions, deletions and substitutions) of the candidate from the true report, which may be up to 26.

	Linear	*Rank*
Grammar, full	0.713	0.696
Grammar, inter2	0.693	0.676
Grammar, empty	0.670	0.651

Table 2: *N*-best list correlations

Table 2 contains linear and rank correlations, averaged over 12 *N*-best lists, with $N = 100$ in each case. Scores clearly tend to be higher for candidates that are closer to the true report, and the correlation improves with the size of the grammar. The improvement is modest, but this may be due to the dominance of low-level paths alluded to previously, and is under investigation.

5.3 Trails

Because of the present formulation of the ARM syntax, words can have several interpretations through singleton reductions. This is leading to a proliferation of trails and is being addressed. The dynamic range of trail scores is quite large, and whether the highest-scoring trails tend to pass through high or low nodes is determined by the grammar structure and the training procedure.

6. CONCLUSIONS

A language model that employs both *n*-grams and grammar rules coherently has a number of potential advantages. It has the robustness of a pure *n*-gram model, an improved capacity to spot meaningful phrases where these extend beyond a local *n*-gram, and the capacity to interpret incoming data. Efficient training and operation is possible using the node-based procedures described in this paper. Training and scoring (including parsing) for the ARM corpus require approx. 1sec and 0.1sec CPU time per sentence respectively, on a top-end UNIX server. Perplexity and *N*-best list re-scoring results are turning out as expected. In particular, enhancing the grammar moves the better candidates up the *N*-best lists.

Some possible refinements to the system include the development of a re-estimation version of the training procedure, the weighting of paths by their structural level, incorporation of extended *n*-grams for word and trail paths, full integration of trigrams throughout the structure, and application to word lattices in preference to *N*-best lists. Of wider significance is the issue of grammatical inference, and it may be of value that this system enables the performance implications of incremental changes to a grammar to be measured.

Acknowledgements

The work is supported by the Speech Research Unit, DRA Malvern, and by the Engineering and Physical Sciences Research Council.

References

[1] G.J.F.Jones, J.H.Wright and E.N.Wrigley, "The HMM interface with hybrid grammar-bigram language models for speech recognition", *Proc. ICSLP-92*, Banff, pp 253-256.

[2] J.H.Wright, G.J.F.Jones and H.Lloyd-Thomas, "A consolidated language model for speech recognition", *Proc. Eurospeech-93*, Berlin, pp 977-980.

[3] J.H.Wright, G.J.F.Jones and H.Lloyd-Thomas, "Language model training and robust parsing for speech recognition", *Proc. Institute of Acoustics (Speech and Hearing Conference)*, vol 16, 1994, pp 63-71.

[4] R.Iyer, M.Ostendorf and J.R.Rohlicek, "Language modelling with sentence-level mixtures", *Proc. ARPA Workshop on Human Language Technology*, Plainsboro, U.S.A., March 1994, pp 82-86.

[5] H.Ney, U.Essen and R.Kneser, "On structuring probabilistic dependencies in stochastic language modelling", *Computer Speech and Language*, vol 8, 1994, pp 1-38.

[6] M.Meteer and J.R.Rohlicek, "Statistical language modelling combining *n*-gram and context-free grammars", *Proc. ICASSP-93*, Minneapolis, pp II-37-40.

[7] S.Seneff, H.Meng and V.Zue, "Language modelling for recognition and understanding using layered bigrams", *Proc. ICSLP-92*, Banff, pp 317-320.

[8] J.H.Wright, G.J.F.Jones and H.Lloyd-Thomas, "Training and application of integrated grammar/bigram language models", in R.C.Carrasco and J.Oncina (Eds), *Grammatical Inference and Applications*, Lecture Notes in Artificial Intelligence vol 862, Springer-Verlag, 1994, pp 246-259.

[9] M.J.Russell, K.M.Ponting, S.M.Peeling, S.R.Browning, J.S.Bridle, R.K.Moore, I.Galiano and P.Howell, "The ARM continuous speech recognition system", *Proc. ICASSP-90*, Albuquerque, pp 69-72.