Proceedings of the Institute of Acoustics

LANGUAGE MODEL TRAINING AND ROBUST PARSING FOR SPEECH RECOGNITION

J.H.Wright[1*], G.J.F.Jones[2] and H.Lloyd-Thomas[3]

[1]Department of Engineering Mathematics, Queens Building, University of Bristol, Bristol
[2]Department of Engineering, University of Cambridge, Trumpington Street, Cambridge
[3]Ensigma Limited, Turing House, Station Road, Chepstow

1 INTRODUCTION

The main objective of this work is to develop enhanced language models for use in speech recognition. In the first instance these models are to be utilised in the re-scoring of N-best lists of candidate sentences from a speech recogniser. The models consist of ordinary and long-distance or "extended" n-grams (bigrams and trigrams), and context-free grammar rules. The extended n-grams are intended to capture remote dependencies between words, that are beyond the scope of conventional n-grams, and word probabilities are dependent on the $n-1$ most informative words within a predefined window. The grammar rules are intended to capture structural features of the sentences, as far as they exist. A full-coverage grammar is not expected (and may not be desirable in practice). Integration of these language model components into a coherent framework is very important, both for training and for scoring, and this is partially achieved (in two respects) in the system described here. First, the extended n-grams are a generalisation of standard n-grams (and reduce to those when the window is closed), and second the grammar rules are invoked as part of a sequential procedure based on bigrams. Full integration would require trigrams (and extended trigrams) in conjunction with grammar rules but this is not yet achieved.

In previous work [1, 2, 3, 4] we have reported results for a hybrid recognition system in which sentences are scored both by a grammar (with assumed full coverage) and by a standard bigram model, with automatic switching between the models. As expected, sentences that obey the grammar are usually scored more highly by that model, with remaining sentences picked up by the bigram model. Sentences are therefore partitioned into two classes, with consequent problems for interpreting and comparing scores for sentences in different classes. The system described in this paper removes this difficulty, and is similar to that described in [5] but is more general in scope. We have also reported work on extended bigrams [6, 7] but here we carry the same idea forward to trigrams. Other researchers [8, 9, 10, 11] have also emphasised the importance of structures wider in scope than conventional trigrams.

2 EXTENDED TRIGRAMS

2.1 Training

Let $C(u, v, w)$ be the count of occurrences of sequences of the form $\ldots u \ldots v \ldots w \ldots$ in training windows, where u, v, w represent words, and w is the first occurrence of this word after v (but can be the same as v). A training window is defined for every word in the training corpus, and extends back from that word either by a fixed number of words or else to the start of the sentence (this is

*Currently at Ensigma Limited, Turing House, Station Road, Chepstow

optional). An extended trigram distribution can be obtained by normalising over w, but this must be smoothed. The extended trigram distribution is therefore defined as

$$P_{etr}(w|u,v) = (1-\lambda_{u,v})\frac{C(u,v,w)}{\sum_w C(u,v,w)} + \lambda_{u,v}P_{ebi}(w|v)$$

where the extended bigram distribution is similarly defined and smoothed using the word unigrams:

$$P_{ebi}(w|v) = (1-\lambda_v)\frac{C(v,w)}{\sum_w C(v,w)} + \lambda_v P_{uni}(w)$$

The smoothing coefficients are assigned using a pseudo-Bayes method described in the Appendix. This optimises the smoothing over the training corpus, and was found to give the best results, notably in comparison with the popular leaving-one-out method.

Extended bigrams and trigrams are stored in a compact tree-based data structure, in parallel with the ordinary bigrams and trigrams, and can be compressed to save space [7].

2.2 Scoring

In order to score a sentence using extended bigrams, for each word in the sentence a single "parent" word is chosen using a relative information criterion, and then a probability assigned to the word from the extended bigram distribution [6, 7]. When the selected parent word is in fact the previous word, the probability assigned is just the standard bigram probability, and this situation is forced if the scoring window (which limits the history in the same way as the training window) is restricted to a single previous word. In this way the extended bigrams generalise upon the conventional bigrams.

Scoring using extended trigrams is the same in principle. Represent the sentence as $\$w_1 w_2 \ldots w_L\$$ where \$ is an end-marker. For each word w_k a scoring window extends backwards from that word, either by a fixed number of words or else to the start of the sentence. Two parent words w_i, w_j with $i < j < k$ are chosen from within this window, by maximising the relative information as follows:

$$\max_{i,j} \left| \log \frac{P_{test}(w_k|w_i,w_j,\bar{\$})}{P_{uni}(w_k)/[1-P_{uni}(\$)]} \right|$$

where

$$P_{test}(w_k|w_i,w_j,\bar{\$}) = \begin{cases} \dfrac{P_{tr}(w_k|w_{k-2},w_{k-1})}{1-P_{tr}(\$|w_{k-2},w_{k-1})} & \text{if } i=k-2,\ j=k-1 \\ P_{norm}(w_k|w_i,w_j) & \text{for } \max\{k-M,1\} \le i < k-2,\ j \ge \max\{t<k : w_t = w_k\} \end{cases}$$

$$P_{norm}(w_k|w_i,w_j) = \frac{P_{etr}(w_k|w_i,w_j)}{1-\sum_{w_t \in \{w_{j+1},\ldots,w_{k-1}\}} P_{etr}(w_t|w_i,w_j)}$$

($P_{tr}()$ is the smoothed standard trigram and M is window length). This finds the two most informative parent words (relative to the word unigram probability). The tested conditional probability assumes that the latest word w_k is not the end-of-sentence marker \$, because a conventional

trigram step is used for this symbol. Note that the standard trigram probability is used for the case where the tested parents are in fact the two preceding words. Otherwise the extended trigram probability is used, but re-normalised to take into account the restriction that w_k must be the first occurrence of this word after the second parent w_j (in conformity with the training count procedure).

The sentence probability is then the product of probabilities of each word given its parent words, with appropriate initial and final standard bigram and trigram steps:

$$P(\$w_1 w_2 \ldots w_L\$) = P_{bi}(w_1|\$)P_{tr}(w_2|\$, w_1)\prod_{k=3}^{L} P_{ext}(w_k|w_{i(k)}, w_{j(k)})P_{tr}(\$|w_{L-1}, w_L)$$

The extended trigram probability is simply the standard trigram probability when the parent words are the two preceding words, otherwise it is derived from the extended trigram distribution:

$$P_{ext}(w_k|w_{i(k)}, w_{j(k)}) = \begin{cases} P_{tr}(w_k|w_{k-2}, w_{k-1}) & \text{if } i(k) = k-2,\ j(k) = k-1 \\ P_{norm}(w_k|w_{i(k)}, w_{j(k)})[1 - P_{tr}(\$|w_{k-2}, w_{k-1}) - P_{corr}(w_{k-2}, w_{k-1})] & \text{otherwise} \end{cases}$$

$$P_{corr}(w_{k-2}, w_{k-1}) = \sum_{w_t \in \{w_{j+1}, \ldots, w_{k-1}\}} P_{tr}(w_t|w_{k-2}, w_{k-1})$$

The (usually small) correction probability $P_{corr}()$ again imposes the restriction that w_k cannot be the same as any other word since w_j, and ensures that the language model is normalised overall. Perplexity comparisons with standard trigram models are therefore valid.

Note that no normalisation over the vocabulary is required at run-time. This makes the speed of the algorithm largely independent of vocabulary size, and although it depends on the length of the scoring window it is very fast if this is reasonably short. Other versions of the procedure have been tested, for example using mixtures of extended trigram distributions instead of choosing two particular parents, but so far the other versions are much slower and the results are generally inferior.

3 GRAMMAR/BIGRAM PATH MODEL

3.1 Training procedure

Suppose now that we have available a set of context-free grammar rules that represent common structures that occur within sentences in the application. This will not in general be a full-coverage grammar, or even close to it, but there are many situations where such rules provide a natural and compact representation that is far superior to that achievable using trigrams. The basic scoring model is still sequential (left-to-right) and the problem is to fit these rules into the model. The main complications are that tree structures can overlap (sharing subtrees that are stored only once for efficiency) and that local ambiguity can exist. See [12] for a discussion of these issues and an introduction to the Tomita generalised LR parsing algorithm.

We use an enhanced version of this algorithm to spot all possible subtrees within a sentence. In general there may be many possible left-to-right paths (through these subtrees) that span the sentence, see for example figure 1.

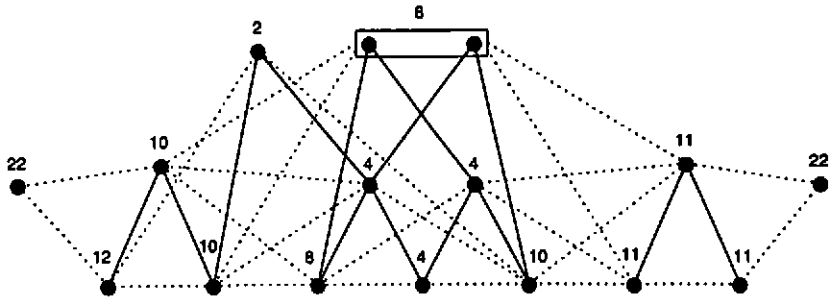

Figure 1: Subtree paths.

The dotted lines are bigram links that join the subtrees. There are 22 paths through the subtrees for this 7-word sentence (including the path through the words themselves), and above each node are shown the numbers of these paths that pass through that node. Similarly (but not marked) there are numbers of paths that pass along each bigram link. These totals are found using a two-pass algorithm [13]. Although the number of paths can become very large (literally billions for long sentences rich in grammar structure) the algorithm is node-based and is therefore quite fast (the current version takes a few seconds per sentence on a multi-user UNIX system).

The proportions of the total paths that use each bigram link are added to corpus totals for the appropriate symbol pairs, and similarly the proportions of paths that pass through each node are accumulated for the appropriate grammar rules (for ambiguous nodes the paths are allocated to the rules in proportion to their numbers of derivations). A special procedure handles null rules. When the training corpus has been covered, bigram counts for a common first symbol are normalised, as are rule counts for a common left-hand-side. After normalisation, symbol bigram probabilities and rule probabilities emerge which reflect their utility in the training corpus. Smoothing is done either by the method in the Appendix or else by the simple expedient of adding a small constant to each count before normalising, and in tests so far the latter gives the best results. Because no re-estimation is involved, only a single pass through the training corpus is required.

3.2 Sentence scoring

Sentence scoring starts from the subtrees as in training, and is based on path scores. Each path score is a product of derivation probabilities (of rules used within subtrees below nodes passed through by the path) and of bigram probabilities (of links used by the path). In this way the top-down grammar scoring and the left-to-right bigram scoring are brought together in a natural way, and the language model is correctly normalised overall.

Let $span(X) = (k_1, k_2)$ denote the part of the sentence spanned by node X where $1 \leq k_1 \leq k_2 \leq L$ for sentence length L. The following (similar to the HMM forward algorithm) finds the overall sentence score as the total path score.

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

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

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

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

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

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

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). If the sentence contains no syntactic structure at all then the score defaults to the straight bigram score.

3.3 Sentence interpretation

Segmenting a sentence into connected syntactic structures provides an interpretation of the sentence, and the score of each interpretation is the sum of scores over all paths bounded above by the particular path (which we call a "trail") through the topmost syntactic nodes. For each sentence there is a (usually small) number of trails that span the sentence at the highest levels (see figure 2). As usual it is the existence of subtree sharing and local ambiguities that complicates the process of finding the best interpretation. The algorithm (described in [13]) is not maximally efficient, and implementation, testing and improvement are in progress.

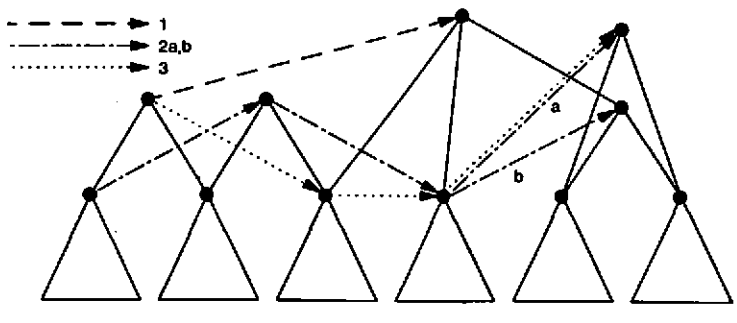

Figure 2: Top-level trails.

4 RESULTS

Results have been obtained for a corpus of Airborne Reconnaissance Mission (ARM) reports [14]. These have a vocabulary of 511 words and each report consists of a series of sentences (overall mean length 9.2 words) of standard types. There is a full grammar for these reports, which we have adapted into context-free form, and now consists of 226 structural rules, in addition to the rules for converting preterminal to terminal symbols. This allows us to study the effect of using a hierarchy of grammars, from very limited to full coverage.

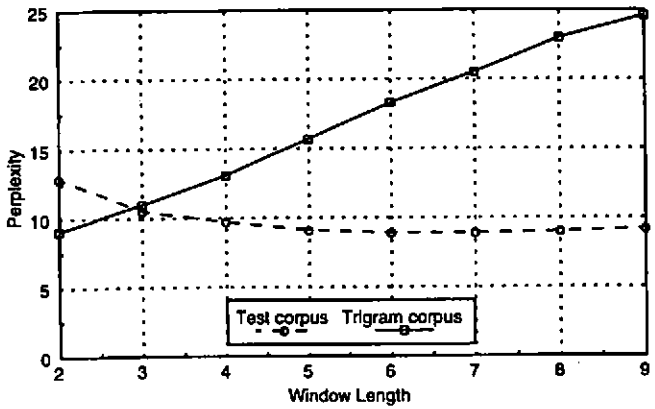

Figure 3: Perplexity results for extended trigrams.

4.1 Corpus perplexity

Figure 3 shows perplexity results for a test corpus of ARM sentences. Because each sentence is distinct there is no purpose in extending the training or scoring windows beyond the start of the current sentence (for example to a cache of recently-used words) so the windows are kept short. A window length of 2 words corresponds to the standard trigram model. A reduction in perplexity of 28% compared with this model is seen for a window length of 6 words. For comparison, the perplexity of a corpus generated at random from the smoothed trigram model rises as the window opens, reflecting the absence of the long-distance correlations that characterise the real data. Table 1 contains perplexity figures for a hierarchy of n-gram models (with window length 6 for extended bigrams and trigrams), and for a hierarchy of grammars, applied to the test corpus and to a corpus generated from each of the smoothed bigram and trigram models. The perplexity 10.0 obtained using the full-coverage grammar compares well with the best results obtained using n-grams. The low perplexity persists as rules are progressively pruned from the grammar. The base-level grammar essentially defaults to a bigram model, but is different from the word bigram model (because of the preterminal to terminal rules which remain, and a different smoothing procedure), hence the difference in perplexity.

Language model	Test corpus perplexity	Bigram corpus perplexity	Trigram corpus perplexity	Word replacement test
Unigram	116.5	106.4	103.9	2.19
Bigram	17.1	16.7	13.4	1.16
Trigram	12.9	25.0	9.1	1.15
Extended bigram	12.3	43.7	19.0	1.14
Extended trigram	8.9	94.8	18.3	1.15
Grammar, full	10.0	24.2	14.9	1.13
Grammar, no top-level	9.5	23.5	14.2	1.11
Grammar, smaller	10.2	25.0	15.2	1.11
Grammar, smaller	9.4	24.9	14.4	1.13
Base-level grammar	32.7	39.7	21.3	1.41

Table 1: ARM perplexity results.

4.2 Word-replacement tests

For each sentence in the test corpus, 10 further sentences are generated by replacing a randomly-chosen word with another (selected at random from the vocabulary), the rank of the score of the correct sentence is found, and hence the mean rank over the corpus. The grammar/bigram system slightly outperforms the n-gram models on this test, as shown by the last column of table 1. This outperformance persists through the pruning of the grammar, and is a positive sign for the application of this procedure to the re-scoring of N-best lists.

We intend to extend these tests to the re-scoring of N-best output sentence hypotheses from the ARM recogniser.

5 CONCLUSIONS

We have described language models that after data-driven training can bring significant reductions in perplexity compared with a standard trigram model. The time penalty with the use of extended trigrams is negligible but the space penalty is not: at present the extended trigrams occupy seven times as much space as the standard trigrams. To alleviate this, substantial data-compression should be possible, however, and this will be investigated as time permits. We have also implemented extended 4-grams but this brought no further reduction in perplexity (probably because of lack of training data) and the space requirements would be prohibitive for a larger vocabulary. The time penalty with the use of the grammar/bigram system is rather larger (although not unrealistic at a few seconds per sentence on average) and there are efficiency gains that should reduce this. The space penalty depends on the size of the grammar.

The extended n-grams could easily be superimposed on the grammar/bigram system at the top level because each trail consists of a unique sequence of nodes. This may improve the capability of the system to find the best interpretation. Extended n-grams could also be applied to the bottom path (through the words). Full integration would be more difficult to achieve in practice because of the large numbers of paths.

The re-scoring of N-best hypotheses is useful for demonstrating knowledge source utility in speech recognisers. However, it is inherently inefficient since much effort is replicated. The language models described here could be incorporated more efficiently by applying them in the processing of word hypotheses lattices. If the models can be demonstrated to be sufficiently useful, the parsing and scoring algorithms used here could be extended to operate in this much more complex implementational environment.

The ARM system has been a useful testbed for these approaches mainly because of the existence of the grammar, but a grammar will not normally be available in future applications, and reductions in perplexity as large as those seen here may not occur when there is greater volatility in sentence construction. We believe that the important benefits will be seen in applications to tasks for which a relatively compact but high utility partial-coverage grammar can be devised. Language models for speech applications have to be driven by (and adapted to) actual data rather than being fabricated *a priori*. By shifting the aim of the grammar away from full coverage and towards the spotting of short meaningful phrases, minimising ambiguity and complementing a high-quality n-gram model instead of attempting to replace it, we hope that grammars will at last be able to make a useful contribution in speech recognition.

6 ACKNOWLEDGEMENTS

We would like to thank the Speech Research Unit, DRA Malvern, for supporting this work, with special thanks to Dr M.J.Russell. The work is also supported by the EPSRC.

REFERENCES

[1] J.H.Wright, G.J.F.Jones, and E.N.Wrigley, "Hybrid grammar-bigram speech recognition system with first-order dependence model," in *Proceedings of ICASSP-92*, (San Francisco), pp. 1-(169-172), IEEE, 1992.

[2] G.J.F.Jones, J.H.Wright, and E.N.Wrigley, "The HMM interface with hybrid grammar-bigram language models for speech recognition," in *Proceedings of the Second International Conference on Spoken Language Processing*, (Banff), pp. 253-256, 1992.

[3] G.J.F.Jones, J.H.Wright, H.Lloyd-Thomas, and E.N.Wrigley, "A hybrid grammar-bigram language model with decoding of multiple (N-best) hypotheses for speech recognition," in *Proceedings of the Institute of Acoustics (Conference on Speech and Hearing)*, (Windermere), pp. 329-336, IOA, 1992.

[4] G.J.F.Jones, *Application of Linguistic Models to Continuous Speech Recognition*. PhD thesis, University of Bristol, 1994.

[5] M.Meteer and J.R.Rohlicek, "Statistical language modelling combining N-gram and context-free grammars," in *Proceedings of ICASSP-93*, (Minneapolis), pp. II-(37-40), IEEE, 1993.

[6] J.H.Wright, G.J.F.Jones, and H.Lloyd-Thomas, "A consolidated language model for speech recognition," in *Proceedings of EUROSPEECH-93*, (Berlin), pp. 977-980, ESCA, 1993.

[7] J.H.Wright, G.J.F.Jones, and H.Lloyd-Thomas, "A robust language model incorporating a substring parser and extended n-grams," in *Proceedings of ICASSP-94*, (Adelaide), pp. I-(361-364), IEEE, 1994.

[8] X.Huang, F.Alleva, H.-W.Hon, M.-Y.Hwang, K.-F.Lee, and R.Rosenfield, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7, pp. 137–148, April 1993.

[9] R.Rosenfeld, "A hybrid approach to adaptive statistical language modelling," in *Proceedings of the ARPA Workshop on Human Language Technology*, (Plainsboro, NJ), pp. 76–81, 1994.

[10] R.Iyer, M.Ostendorf, and J.R.Rohlicek, "Language modelling with sentence-level mixtures," in *Proceedings of the ARPA Workshop on Human Language Technology*, (Plainsboro, NJ), pp. 82–86, 1994.

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

[12] M.Tomita, *Efficient parsing for Natural Language*. Kulwer Academic Publishers, 1986.

[13] J.H.Wright, G.J.F.Jones, and H.Lloyd-Thomas, "Training and application of integrated grammar/bigram language models," in *Proceedings of the Second International Colloquium on Grammatical Inference*, (Alicante), 1994.

[14] 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," in *Proceedings of ICASSP-90*, (Albuquerque), pp. 69–72, IEEE, 1990.

[15] Y.M.Bishop, S.E.Feinberg, and P.W.Holland, *Discrete Multivariate Analysis: Theory and Practice*. M.I.T. Press, 1975.

APPENDIX: Pseudo-Bayes Optimal Smoothing

If $C_k(k = 1, \ldots, K)$ is the observed frequency of the kth event in a total of N, so the maximum likelihood estimate of the true probability p_k is given by C_k/N, then the smoothed estimate of p_k is given by

$$\hat{p}_k = (1 - \lambda)\frac{C_k}{N} + \lambda r_k$$

where r_k is an independent estimate of p_k (such that $\sum_{k=1}^{K} r_k = 1$), and

$$\lambda = \frac{M}{N + M} \quad \text{where} \quad M = \frac{N^2 - \sum_{k=1}^{K} C_k^2}{\sum_{k=1}^{K} [C_k - N r_k]^2}$$

It can be shown [15] that using this value of λ minimises the mean square error $E[\sum_{k=1}^{K} (\hat{p}_k - p_k)^2]$, provided the counts$C_k(k = 1, \ldots, K)$ follow a multinomial distribution. In this way, unigrams are used to smooth the bigrams (ordinary and extended), and these are respectively used to smooth the trigrams (ordinary and extended).