

LANGUAGE MODEL REPRESENTATIONS FOR BEAM-SEARCH DECODING

Giuliano Antoniol, Fabio Brugnara, Mauro Cettolo and Marcello Federico

IRST-Istituto per la Ricerca Scientifica e Tecnologica

I-38050 Povo (Trento), Italy

{antoniol,brugnara,cettolo,federico}@irst.it

ABSTRACT

This paper presents an efficient way of representing a bigram language model for a beam-search based, continuous speech, large vocabulary HMM recognizer. The tree-based topology considered takes advantage of a factorization of the bigram probability derived from the bigram interpolation scheme, and of a tree organization of all the words that can follow a given one. Moreover, an optimization algorithm is used to considerably reduce the space requirements of the language model. Experimental results are provided for two 10,000-word dictation tasks: radiological reporting (perplexity 27) and newspaper dictation (perplexity 120). In the former domain 93% word accuracy is achieved with real-time response and 23 Mb process space. In the newspaper dictation domain, 88.1% word accuracy is achieved with 1.41 real-time response and 38 Mb process space. All recognition tests were performed on an HP-735 workstation.

1. INTRODUCTION

Many current ASR systems generate initial hypotheses through a beam-search decoding algorithm that employs a Finite State Network (FSN) representing a bigram Language Model (LM). Successive refinements can be performed on smaller search spaces by applying more powerful LMs. Nevertheless, efficiently managing bigram LMs is a crucial issue in ASR.

This paper addresses the problem of statically representing a bigram LM with a FSN to be used by a Viterbi based beam-search HMM recognizer. A tree-based representation is described and a reduction technique is proposed to sensibly lower its size. Experiments are reported for two continuous speech dictation tasks: a radiological reporting application, called A.Re.S. [3], and a newspaper dictation application. Both lexicons consist of about 10,000 Italian words.

In the following, the interpolated LM estimation technique is outlined and the tree-based LM representation is described. Finally, experimental results are

provided and discussed

2. LM ESTIMATION

Two basic computation schemes of bigrams are generally employed: backing-off and interpolation [4]. The interpolation scheme:

$$Pr(z | y) = f'(z | y) + \lambda(y)Pr(z) \quad (1)$$

is preferable because it allows an efficient representation of the search space. The interpolation scheme requires estimating a discounted relative frequency $f'(z | y) \leq f(z | y)$ and a probability $0 \leq \lambda(y) \leq 1$ for the words that never occurred in context y . For this estimation problem several techniques can be applied. In a previous work [4] it was shown that the *Stacked* estimation algorithm for the general linear discounting model:

$$f'(z | y) \triangleq (1 - \lambda(y))f(z | y)$$

favorably compares with the best ones.

3. LM REPRESENTATION

A static representation of the search space is attractive because there is no overhead in building it during the recognition process, and because some network optimization can be performed off-line.

3.1. Linear Representation

An interpolated bigram LM could be implemented with the explicit representation of all the possible links between word pairs. Placeway et al. [9] showed how to use explicit links only between word pairs for which $f'(z | y) > 0$, by using a *null node* for the other events (see Figure 1). If $|V|$ is the vocabulary size and d the number of different observed bigrams, at most $d + 2|V|$ links connecting words are necessary if the null node is used. For non trivial vocabularies this number is definitely smaller than the $|V|^2$ links required by the fully connected network.

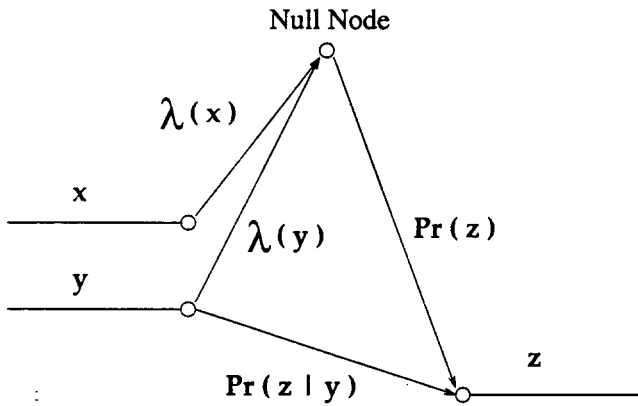


Figure 1: LM representation with the *null node*.

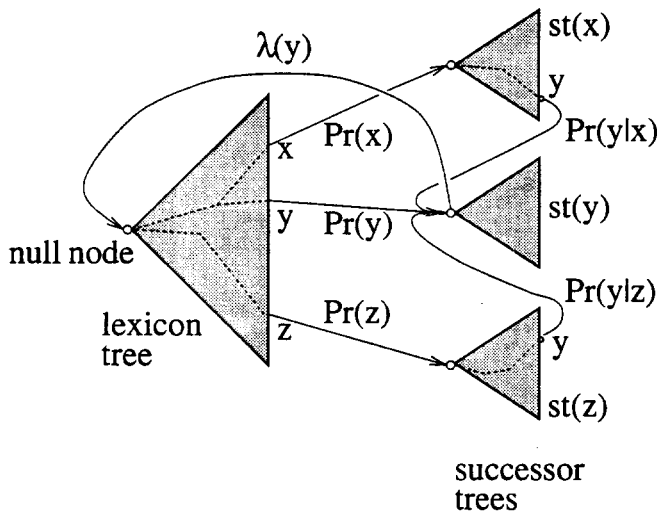


Figure 2: Static Tree-based Representation.

3.2. Tree-based Representation

The acoustic “similarity” of words is not taken into account in the linear representation. As a matter of fact, in a large vocabulary many words share the initial portion of their phonetic transcription. Ney et al. [7, 6] have shown that advantages can be obtained by integrating a tree organization of the lexicon with the beam-search algorithm. Unfortunately, in a lexical tree, the identity of a word is only known at the leaf level: hence, in order to integrate the bigram probability, a duplicate of the whole lexicon would be necessary for each word. In the following, a method of coping with this problem is presented, as an alternative to the already known ones - e.g. dynamic construction of the search space [7, 8], or static linear-tree mixed network topology [5].

The representation considered here combines the null node idea with that of lexical trees. For each word

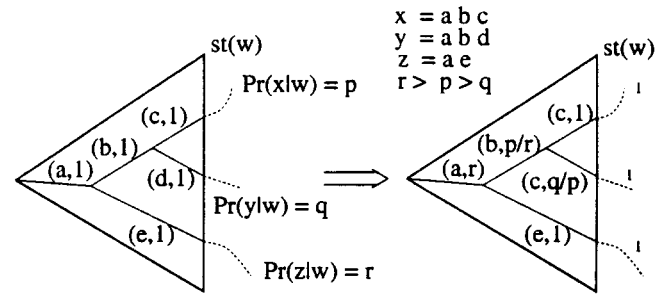


Figure 3: Factorization of probabilities in a tree.

x , the set of successor words for which $f'(y | x) > 0$ is organized as a tree. The proposed FSN structure is depicted in Figure 2. The leftmost triangle represents the whole lexicon tree, while the small triangles represent successor trees ($st(\cdot)$). The static representation of the tree-based network is effective since the average size of the successor trees is usually much smaller than the size of the whole lexicon. Moreover, successor trees may also be reduced by considering a subset of bigrams as explained in [5].

3.3. Factorization of Probabilities

Acoustic information, related to phoneme HMMs within trees, and linguistic information, specified by the bigram probabilities of arcs among trees, are placed in different regions of the network. In order to use linguistic information as soon as possible during the beam-search, bigram probabilities can be *factorized* [8]. In a lexical tree, when more words share some part of their transcription, the maximum of their probabilities can in fact be propagated toward the root (see Figure 3).

3.4. Network Reduction

After factorizing probabilities, empty arcs outgoing from leaves with probability equal to one can be eliminated by collapsing the states linked by them. Moreover, many arcs of word endings (Figure 3) have probability equal to one, which means that there are redundant paths that can be merged.

The network can be optimized by one of the several known algorithms for minimizing the number of states of a deterministic finite state automaton. The partitioning algorithm described in [1] can be used for this purpose. Its time complexity is $O(mn \log n)$, where m is the number of different symbols labeling arcs, and n the number of states of the network. It should be noticed that symbols of network in Figure 3 are pairs (*phoneme*, *probability*) whose total number can

be comparable with n . Therefore, a direct application of the above mentioned algorithm can be computationally very expensive. Work is under way to devise alternative reduction methods.

4. EXPERIMENTS

Experiments were performed on two 10,000-word dictation domains: radiological reports (A.Re.S.), and articles from the Italian financial newspaper "Il Sole 24 Ore" (S240). The A.Re.S. LM was trained on a 2Mw corpus consisting of all the reports produced in three years by a radiology department. The S240 LM was trained on a 25Mw corpus corresponding to all the issues of year 1990. In both cases the lexicon consisted of the 10,000 most frequent words.

The two lexicons were phonetically transcribed with a set of 50 context independent units. The corresponding HMMs have left-to-right topologies of three or four states, and continuous mixture densities with a number of gaussian components varying from 10 to 24, resulting from an automatic selection process. Acoustic models were trained on the APASCI acoustic database [2]. Sentences and speakers of this training set are different from those in the recognition experiments. This is also true for the sentences used to train the LMs. A model for silence was also trained and included in each LM representation. The acoustic front-end and the decoding algorithm are described in [4]. Recognition tests were carried out on an HP-735 workstation. Memory consumption corresponds to the peak process size measured during decoding. This value is related to the duration of the input utterances. For both tasks the maximum length is about 30 seconds.

4.1. AReS Task

4.1.1. Test Material

Four physicians were asked to read different lists of reports. Recordings were performed in a quiet room. In total, 759 reports were acquired amounting to 4^h44' of speech.

4.1.2. Network Compiling

In order to compare different LM topologies, the Stacked bigram LM [4] was trained on all the available material but the 759 reports. The resulting test set perplexity was 27. The LM was mapped into three different networks: linear, tree-based and optimized. The lexicon tree required 43,600 arcs. The average number of arcs in the successor trees was 73, and the final network had about 840,000 arcs. The network to be

optimized had about 136,000 different (*phoneme, probability*) symbols. The sizes of the three networks are reported in Table 1. Labeled and empty arcs are counted separately, since they present a different computational cost during the speech decoding process.

Topology	#States	#Full Arcs	#Empty Arcs
Linear	134,439	124,002	165,478
Tree-based	687,329	821,626	20,744
Optimized	195,012	311,129	15,659

Table 1: A.Re.S. task: network sizes.

The reduction provided by the optimization step is remarkable: the number of full arcs became about three times smaller.

4.1.3. Recognition Tests

Results reported in Table 2 show that the tree-based representation outperforms the linear one in terms of word accuracy (WA) and real-time ratio response (RTR, i.e. recognition-time/speech-duration). Because of the higher average number of active arcs per frame (#Arcs/Frame in Table 2), recognition time with the linear FSN is 5 times higher than with the tree-based one, despite the former net being considerably smaller. For the same reason, the linear net requires more dynamic memory.

As Table 2 shows, the reduced FSN performs as well as the tree-based one, both in terms of accuracy and speed, but requires less than half the memory.

Topology	#Arcs/Frame	WA	RTR	Process Size
Linear	2350	90.8%	4.93	70 Mb
Tree-Based	292	93.0%	1.01	58 Mb
Optimized	285	93.0%	1.01	23 Mb

Table 2: A.Re.S. task: recognition tests.

4.2. S240 Task

4.2.1. Test Material

Test sentences were randomly extracted from issues of one month. Six female and six male speakers were asked to read 30 sentences each, with verbalized punctuation. Recordings were performed in an office environment.

4.2.2. Network Compiling

Two language models [4] were tested: the Stacked LM (St), and the Shift-1 (S1) LM, which provided a good trade-off between number of bigrams to be represented (732,362 vs. 1,462,929) and perplexity (120 vs. 117). Sizes of the three considered LM representations are reported in Table 3. The lexicon tree requires about 30,000 arcs. The reduced tree-based FSN was generated by using a sub-optimal version of the algorithm mentioned in Section 3.4, which results more convenient both in terms of time and space requirements. Moreover, the achieved compression factor was comparable with that of the optimal algorithm.

LM	Top.	#States	#Full Arcs	#Empty Arcs
S1	Lin.	109,732	99,686	744,331
"	Tree	2,601,650	3,315,828	20,101
"	Red.	603,729	1,317,808	20,101
St	Lin.	109,732	99,686	1,470,131
"	Tree	5,415,982	6,855,961	20,101
"	Red.	1,145,070	2,585,049	20,101

Table 3: *S24O task: network sizes.*

4.2.3. Recognition Tests

Recognition performance was evaluated only on the reduced nets. Results are given in Table 4. The network of the Shift-1 LM performs slightly worse, but requires much less memory.

LM	PP	#Arcs/ Frame	WA	RTR	Process Size
St	117	805	88.5%	1.46	66Mb
S1	120	767	88.1%	1.41	38Mb

Table 4: *S24O task: recognition tests.*

5. CONCLUSIONS AND FUTURE WORK

A technique for representing bigram language models has been presented. The proposed method incorporates the null node idea for compact bigram representation, the lexical tree structure to improve beam-search decoding, and a reduction step to overcome the problem of high space requirements raised by the lexical tree approach. Experiments on newspaper dictation show

that this technique is viable for large vocabulary and high perplexity applications. Future work will be dedicated to further refine the current optimization algorithm by exploiting knowledge about the structure of the tree-based network.

6. REFERENCES

- [1] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [2] B. Angelini, G. Antoniol, F. Brugnara, M. Cettolo, M. Federico, R. Fiutem, and G. Lazzari. Speaker independent continuous speech recognition using an acoustic-phonetic Italian corpus. In *Proceedings of the ICSLP*, Yokohama, Japan, September 1994.
- [3] B. Angelini, G. Antoniol, F. Brugnara, M. Cettolo, M. Federico, R. Fiutem, and G. Lazzari. Radiological reporting by speech recognition: the A.Re.S system. In *Proceedings of the ICSLP*, Yokohama, Japan, September 1994.
- [4] G. Antoniol, F. Brugnara, M. Cettolo, M. Federico. Language model estimations and representations for real-time continuous speech recognition. In *Proceedings of the ICSLP*, Yokohama, Japan, September 1994.
- [5] H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger. Techniques to achieve an accurate real-time large-vocabulary speech recognition system. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ, 1994.
- [6] H. Ney. Architecture and search strategies for large-vocabulary continuous-speech recognition. *New Advances and Trends in Speech Recognition and Coding*. Springer-Verlag, 1993.
- [7] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. In *Proceedings of the IEEE ICASSP*, S. Francisco, CA, 1992.
- [8] J. Odell, V. Valtchev, P. Woodland, and S. Young. A one pass decoder design for large vocabulary recognition. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ, 1994.
- [9] P. Placeway, R. Schwartz, P. Fung, and L. Nguyen. The estimation of powerful language models from small and large corpora. In *Proceedings of the IEEE ICASSP*, Minneapolis, MN, 1993.