

the linguistic randomly chosen test set. Recognition results confirm the difference between the naive methods, A1 and S1, and the refined ones, which exhibit substantially equivalent performance. The ranking among the best models is probably influenced by the nature of the testing data.

	clfu	enmo	euma	lube	avg.
A1	81.48	90.28	90.13	84.91	86.70
GT	89.82	94.73	94.48	92.16	92.80
S1	88.91	94.13	93.87	91.15	92.01
S β	89.85	94.65	94.59	92.02	92.78
LE	89.97	94.67	94.98	92.66	93.07
LS	89.48	94.77	93.97	92.06	92.57
LGL	89.68	94.68	94.59	92.75	92.93
LGS	89.70	94.70	94.88	92.62	92.97

Table 4: *LM recogniton performance with different speakers in terms of WA.*

Recognition tests (Table 5) were also performed with the two different search space organizations and with the tree-based one after the reduction step. Both acoustic framework and LM (LG Stacked) were kept fixed, and the beam threshold was chosen to achieve real-time response on a HP735 workstation.

Word accuracy obtained with the linear representation is slightly than with the tree-based one, given the different impact of the beam threshold on the two topologies. Moreover, due to the higher average number of hypotheses per frame, the recognition is 5 times slower and the dynamic process size is larger despite the fact that the linear network be the smallest one. Finally, the network reduction does not affect recognition performance, but has a great influence on the process memory requirements, as expected.

Computational advantages of the tree-based representation vs the linear one can be seen in Figure 2. In this figure the evolution of the number of active arcs during the decoding of a speech segment is shown, on linear and logarithmic scales. The example reflects a qualitative behavior which was observed in many cases, confirming that the most ambiguous regions are between-word transitions, either actual or potential. The peaks of ambiguity are much more severe in the case of the linear net. In fact, the relevant difference between the two representations is confined to these regions, since, as is evident from the log plot, in the word-ending regions the number of active arcs can happen to be lower for the linear net. The within-word peaks correspond to potential word boundaries. Notice for example the behavior on the word “persistenza”,

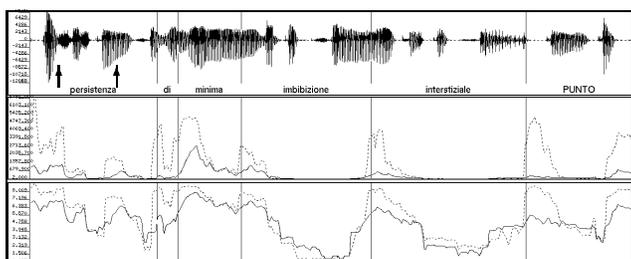


Figure 2: *Number of active arcs during decoding of a speech segment on linear and log scales with linear (dashed lines) and tree-based (solid lines) topologies.*

Table 5: *Comparison among linear, tree-based and reduced tree-based representations in terms of size, WA, real-time ratio and recognition process size.*

whose beginning parts “per” and “persiste” are words by themselves: two peaks arise in correspondence with their ending times, marked by arrows.

V. CONCLUSIONS

Several bigram LM estimations were evaluated in terms of perplexity on different corpora, exhibiting small but significant differences. The LMs were also compared in terms of word accuracy after integration in a 10,000-word continuous speech recognizer, showing that refined models give comparable results. Furthermore, the advantage of a tree-based LM representation on the beam-search algorithm was discussed. Finally, an off-line reduction of the static tree-based network was proposed as a viable method to overcome the problem of memory size.

References

- [1] 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. *ICSLP*, 1994.
- [2] B. Angelini, F. Brugnara, D. Falavigna, D. Giuliani, R. Gretter and M. Omologo. Speaker independent continuous speech recognition using an acoustic-phonetic Italian corpus. *ICSLP*, 1994.
- [3] A. Aho, J. Hopcroft, J. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [4] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov process. *Inequalities 3*, 1972.
- [5] M. Federico. Stacked estimation of interpolated n-gram language models. *IEEE Workshop ASR*, 1993.
- [6] F. Jelinek, R. L. Mercer, and S. Roukos. Principles of lexical language modeling for speech recognition. In S. Furui and M. M. Sondhi, ed., *Advances in Speech Signal Processing*, pp 651–699. M. Dekker, Inc., 1992.
- [7] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. ASSP-35*(3):400–401, 1987.
- [8] H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger. Techniques to achieve an accurate real-time large-vocabulary speech recognition system. *Proc. ARPA HLT Workshop*, pp 368-373, 1994.
- [9] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.
- [10] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. *Proc. ICASSP*, I: 9–12, 1992.
- [11] J. Odell, V. Valtchev, P. Woodland, and S. Young. A one pass decoder design for large vocabulary recognition. *Proc. ARPA HLT Workshop*, pp 380–385, 1994.
- [12] P. Placeway, R. Schwartz, P. Fung, and L. Nguyen. The estimation of powerful language models from small and large corpora. *Proc. ICASSP*, II:33–36, 1993.
- [13] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. IT-37*(4):1085–1094, 1991.

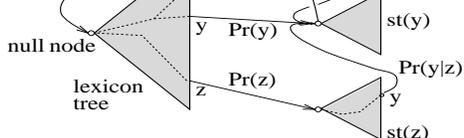


Figure 1: *Tree-based LM representation.* $st(w)$ indicates the tree of successors of word w .

is impossible: for the 10,000-word A.Re.S. task, such a network would need 500×10^6 phoneme-labelled links!

However, the computational efficiency provided by the tree-based representation justifies the efforts made by many research laboratories to overcome the problem of its memory space requirements. In fact, some labs dynamically build the portion of the currently explored space [10, 11]; others adopt a static linear-tree mixture approach [8].

As a matter of fact, a static representation of the whole search space is attractive mainly for two reasons: first, there is no overhead in building it during the recognition process; secondly, the network can be reduced off-line.

In fact, if the null node factorization is also used for the tree-based representation (see Figure 1), for each word only the successors seen in the training text have to be tree-organized. This makes a static tree-based representation effective also for non-naive tasks, since the average size of successor trees is usually small.

One further step which gives computational improvements to the beam-search is the use of the upper-bound among probabilities of words that share a phoneme. This “back-propagation” of probabilities within trees makes many paths redundant, a fact that can be exploited to reduce network size. The partitioning algorithm described in [3] was successfully used for this purpose.

IV. EXPERIMENTS

4.1 System Description.

Acoustic modelling uses phonetic transcription of words with 50 context independent units. Unit HMMs have simple left-to-right topologies of three or four states, depending on the average length of the corresponding units. Distributions are gaussian mixtures with a variable number of components, resulting from a training process which initializes all mixtures with 24 components, and then prunes less used gaussians. The final configuration used in the experiments reported includes a total of 2863 gaussians grouped in 281 mixtures. The signal processing front-end provides the recognizer with a 27-dimensional vector every 10ms, consisting of 8 MEL scaled cepstral coefficients, the log-energy, and their first and second time derivatives. The acoustic parameter vector is scaled so as to ensure that all its elements have comparable ranges. In evaluating gaussian mixtures, instead of summing all the terms, the approximation of taking the most likely term is done, since this allows a time gain without affecting accuracy.

Acoustic models were trained with MLE on a set of 2000 sentences belonging to a phonetically rich database under collection at IRST [2]. Neither the sentences nor the speakers in the training set have any relation with the application domain.

Table 2: *Statistics of the used corpora.* The coverage rate gives the percentage of bigrams in the test which have occurred at least once in the training data.

Model	Concierge	AreS	LOB
A1	57.58 ₈	69.29 ₈	791 ₈
GT	20.83 ₄	19.38 ₂	422 ₂
S1	23.05 ₇	21.11 ₇	472 ₇
S β	20.47 ₁	19.35 ₁	421 ₁
LE	20.53 ₂	19.48 ₃	443 ₅
LS	21.63 ₆	20.41 ₆	465 ₆
LG LOO	21.46 ₅	19.65 ₅	441 ₄
LG Stacked	20.77 ₃	19.51 ₄	431 ₃
O(STD)	$E - 05$	$E - 06$	$E - 05$

Table 3: *Perplexity measures on different texts, LM ranking and order of magnitude of the standard deviation.*

The implementation of the decoding algorithm takes into account that the network can be huge. Hence, in spite of the network being statically represented, the memory used in intermediate computations is allocated on demand with a simple caching strategy. In this way the memory occupancy becomes strictly related to the decoding process speed, for which the most important factor is not the full network size, but the number of expanded arcs. Moreover, caching of distribution values is performed, ensuring that every distribution is computed at most once in every frame, even if the same model appears on many arcs.

4.2 LM Comparison

The perplexity of each LM was evaluated on three text corpora presenting different data sparseness conditions (see Table 2). Testing and training data were randomly extracted in 1:4 proportion. Performance and LM rank list on each task are reported in Table 3. Even if measured perplexities are often very close to each other, the estimated standard deviations are so small that, with high confidence, the differences are real and not random.

The results mainly show a significant difference between the two naive methods, A1 and S1, and the other more refined methods. Interestingly, the LM best performing is the S β method which shows the highest robustness against data sparseness and distribution noise. In fact, the GT-based LM, which performs very well on the large data corpora, falls slightly behind on the smallest corpus where the frequency distribution d_i is probably not well approximated by countings. LE performs well on the small corpus but worsens a little as the corpus size increases. The LG Stacked model performs worse than, but close to, the S β one and shows good robustness. Finally, the LG Stacked model significantly surpasses the LG LOO one.

4.3 Recognition Tests

All LMs were evaluated on the 10,000-word A.Re.S. task in terms of recognition accuracy. Test were performed on a set of 759 radiological reports, amounting to 4 hours and 44 minutes of speech, recorded by 4 physicians, only one of who (*enmo*) had some experi-

	$h(yz) = \frac{d_{c(yz)+1}}{d_{c(yz)}} \frac{c(yz)+1}{c(yz)}$	$\sum_z f'(z y)$
S1	$\max \left\{ \frac{c(yz)-1}{c(y)}, 0 \right\}$	$\frac{d(y_-)}{c(y)}$
S β	$\max \left\{ \frac{c(yz)-\beta}{c(y)}, 0 \right\}$	$\beta \frac{d(y_-)}{c(y)}$
	$\beta \approx \frac{d_1}{d_1+2d_2} < 1.$	
LE	$\frac{c(yz)}{c(y)+d(y_-)}$	$\frac{d(y_-)}{c(y)+d(y_-)}$
LS	$(1-\alpha)f(z y)$	$\alpha = \frac{d_1}{c}$
LG	$(1-\lambda(y))f(z y)$	$\lambda(y)$

Table 1: Estimators for the discounted frequency function $f'(z|y)$ and the zero-frequency probability $\lambda(y)$.

and K_y is an appropriate normalization constant such that:

$$\sum_z Pr(z|y) = 1$$

Interpolation scheme. The bigram probability is computed as a weighted sum of the discounted frequency and the redistributed zero-frequency probability:

$$Pr(z|y) = \begin{cases} f'(z|y) + \lambda(y)Pr(z) & \text{if } c(y) > 0 \\ Pr(z) & \text{if } c(y) = 0 \end{cases} \quad (2)$$

From the point of view of performance and implementation of speech recognition LMs, the interpolation scheme turns out to be preferable. In fact, if similar results can be obtained for both schemes (in terms of perplexity), very efficient LM representations are possible for the latter model (see next section).

2.2 Discounting methods

Several frequency discounting methods as well as zero-frequency estimators have been proposed in the literature. In Table 1 and in the following paragraphs a compendium of the most known approaches is given.

Adding-1 (A1). This very simple estimator derives from the Bayesian estimation criterion.

Good-Turing (GT) formula. It was first introduced by Katz [7] for backing-off n-gram estimation. In the same way suggested in [7], the discounting function h was actually modified such that $h(yz) = 1$ for $c(yz) > 5$.

Absolute (or “shift”) discounting. A small constant β is subtracted from all bigram counts. Both the simplest solution with $\beta = 1$ (S1) ³ [13] and the one proposed by Ney *et al.* [9] for $0 < \beta < 1$ (S β) are considered.

Linear discounting. Empirical frequencies are discounted in proportion to their value. The Linear Empirical (LE) discounting method was described by Witten and Bell [13] and was first employed for LM estimation by Placeway *et al.* [12]. The basic idea is to

³Condition $c(yz) > 0$ in (1) becomes $c(yz) > 1$.

start which can be estimated with the GT formula. Finally, the Linear General (LG) method introduces $|V|$ parameters whose estimation in case of an interpolated model (2) will be described in the next subsection.

2.3 Linear Interpolation Estimation

In the LG interpolated LM the bigram probability is expressed as follows:

$$Pr(z|y) = (1-\lambda(y))f(z|y) + \lambda(y)Pr(z) \quad (3)$$

where $0 < \lambda(y) \leq 1 \forall y$ and $\lambda(y) = 1$ if $c(y) = 0$. Parameter estimation of this model from a training text W is well known in the literature of LMs and HMMs [6]. In fact, the following Leaving-One-Out (LOO) iterative formula derived from Baum-Egon’s estimator [4] was devised:

$$\lambda^{n+1}(y) = \frac{1}{|S_y|} \sum_{yz \in S_y} \frac{\lambda^n(y)Pr(z)}{(1-\lambda^n(y))f^*(z|y) + \lambda^n(y)Pr(z)}$$

where S_y is the set of all occurrences of bigrams of type y_- in W and $f^*(z|y)$ is the relative frequency computed on W after deleting an occurrence of yz .

Stacked Estimation. In order to avoid overtraining, iterations on single parameters can be stopped as soon as their values becomes stable (LG LOO estimation), or their performance on a random cross-validation sample worsens. This latter methods actually provided better results. Further, a way to reduce the disadvantage of deleting a cross-validation set was introduced by using a *Stacked* version of the interpolation model (LG Stacked) [5]. The basic idea of the stacked method is to combine parameters estimated on different random partitions of the training data (into training and cross-validation sets) in order to improve performance.

III. LM REPRESENTATION

The general framework considers search performed by a Viterbi-based beam-search algorithm on a finite state network. Words are represented as sequences of phonetic units modelled with HMMs.

In the straightforward *linear* approach in which each word pair is linked by an arc with associated LM probability, the network size grows as $|V|^2$. Fortunately, the interpolation scheme allows a compact representation which only requires links between word pairs that occurred in the training data and factorizes the bigrams never seen by using a null node [12].

It is typical that within a medium-large vocabulary many words share the initial portion of their phoneme transcription. This suggests that the lexicon be organized as a *tree* in which common beginning phonemes of words are shared and each leaf corresponds to a word. Further, computational advantages obtained by integrating this lexicon representation with the beam-search algorithm are well known [10].

Unfortunately, unlike the linear representation, in the lexicon tree the identity of a word is only known at the leaf level: so, to integrate the bigram probability, a duplicate of the whole lexicon is necessary for each word and the LM probability is applied at the end of the second word of each pair. This implies an increase in the static memory space required by the tree organization to the extent that its “naive” implementation

Giuliano Antoniol, Fabio Brugnara, Mauro Cettolo and Marcello Federico

*IRST-Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Povo (Trento), Italy*

ABSTRACT

This paper compares different ways of estimating bigram language models and of representing them in a finite state network used by a beam-search based, continuous speech, and speaker independent HMM recognizer. Attention is focused on the n-gram interpolation scheme for which seven models are considered. Among them, the Stacked estimated linear interpolated model favourably compares with the best known ones. Further, two different static representations of the search space are investigated: "linear" and "tree-based". Results show that the latter topology is better suited to the beam-search algorithm. Moreover, this representation can be reduced by a network optimization technique, which allows the dynamic size of the recognition process to be decreased by 60%. Extensive recognition experiments on a 10,000-word dictation task with four speakers are described in which an average word accuracy of 93% is achieved with real-time response.

I. INTRODUCTION

This paper compares different ways of estimating bigram Language Models (LMs) and of representing them in a Finite State Network (FSN) used by a beam-search based, continuous speech, and speaker independent HMM recognizer. Within the interpolation scheme seven bigram LMs in the literature are introduced. Comparisons are performed on text corpora presenting increasing data sparseness and on a 10,000-word speech recognition task from the A.Re.S. [1] (Automatic REporting by Speech) applicative domain¹. If better bigram estimates can improve the search engine accuracy, a suitable organization of the search space can improve its speed as well. Because search is performed with a Viterbi based beam-search on FSNs, two different but equivalent (i.e. preserving the same LM) topologies of the search space were investigated: *linear* and *tree-based*. Results show that the latter representation is better suited to the beam-search algorithm as it outperforms the linear one in terms of speed by almost 5 times without affecting recognition accuracy, which is around 93%. Finally, an off-line reduction on the tree-based topology is applied which significantly reduces space requirements of the recognition process.

¹A.Re.S. is a real-time ASR system for radiological reporting developed at IRST in collaboration with the Radiological Department of S. Chiara Hospital, Trento.

Notation

V	vocabulary set
yz	bigram
$y-$	any bigram starting with x
$c(\cdot)$	number of occurrences in a text
$d(\cdot)$	number of different occurrences
$d_i(\cdot)$	number of different i -time occurrences
c, d_i	number of bigram occurrences and different i -time occurrences

II. LM ESTIMATION

2.1 Basic bigram schemes

The LMs treated here² require the estimation of the basic bigram probability: $Pr(z | y)$ from a training sample W . In general, the above probability is computed by combining two components: a discounting function and a redistribution function. The first function is related to the zero-frequency estimation problem[13]: that is, a probability for all the bigrams never occurred in W is computed by discounting the bigram relative frequency $f(z | y) = \frac{c(yz)}{c(y)}$. The second function redistributes the zero-frequency probability among the never seen bigrams. In general, probability is redistributed according either to a less specific distribution - e.g. the bigram distribution if trigrams are computed - or otherwise (e.g. for unigrams) uniformly. The discounting and the redistribution functions are generally combined according to two main schemes: *backing-off* [7] and *interpolation* [6].

Backing-off scheme. Bigram probability is computed by choosing the most significant approximation according to the frequency countings:

$$Pr(z | y) = \begin{cases} f'(z | y) & \text{if } c(yz) > 0 \\ K_y \lambda(y) Pr(z) & \text{if } c(yz) = 0 \wedge c(y) > 0 \\ Pr(z) & \text{if } c(y) = 0 \end{cases} \quad (1)$$

where f' denotes the discounted frequency distribution, $\lambda(y)$ the zero-frequency probability, satisfying $\forall y, z$ such that $c(y) > 0$:

$$0 \leq f'(z | y) \leq f(z | y)$$

$$\sum_z f'(z | y) = 1 - \lambda(y) \leq 1$$

²The following bigram schemes and estimation methods can be recursively extended to higher-order n-grams.