Published in IEEE Workshop on Spontaneous Speech Processing and Recognition, April 14-16, 2003, Tokyo

USING CONTINUOUS SPACE LANGUAGE MODELS FOR CONVERSATIONAL SPEECH RECOGNITION

Holger Schwenk and Jean-Luc Gauvain

LIMSI-CNRS
91403 Orsay cedex, bat. 508, B.P. 133, FRANCE
{schwenk,gauvain}@limsi.fr

ABSTRACT

Language modeling for conversational speech suffers from the limited amount of available adequate training data. This paper describes a new approach that performs the estimation of the language model probabilities in a continuous space, allowing by these means smooth interpolation of unobserved n-grams. This continuous space language model is used during the last decoding pass of a state-of-the-art conversational telephone speech recognizer to rescore word lattices. For this type of speech data, it achieves consistent word error reductions of more than 0.4% compared to a carefully tuned backoff n-gram language model.

1. INTRODUCTION

Conversational speech recognition is known to be a significantly more difficult task than recognition of broadcast news (BN) data. Based on the NIST speech recognition benchmarks [1], current best BN transcription systems achieve word error rates around 15% in 10xRT while the word error rate for the DARPA conversational telephone speech recognition task is about 25% using much more computational resources (100–300xRT). A large amount of this difference can of course be attributed to the difficulties in acoustic modeling, but language modeling of conversational speech also faces problems that are much less frequent in BN data such as unconstrained speaking style, frequent grammatical errors, hesitations, start-overs, etc. In addition, language modeling for conversational speech suffers from an extreme lack of adequate training data since the main data source is audio transcriptions, in contrast to the BN task for which other news sources are readily available. If we consider for instance the DARPA SWB task, there are only about 3M words of transcriptions corresponding to the 260h of available transcribed acoustic training data. Unfortunately, collecting large amounts of conversational LM data is very costly. One possibility is to increase the amount of training data by selecting conversational like sentences in BN material or by transforming other sources to be more conversational-like, see for instance [2, 3]. In this paper, we focus on a language modeling technique that makes better use of the limited amount of data than conventional backoff n-gram models.

In standard backoff n-gram language models words are represented in a discrete space, the vocabulary. This prevents “true interpolation” of the probabilities of unseen n-grams since a change in this word space can result in an arbitrary change of the n-gram probability. The most prominent technique is to backoff to lower order n-grams and word class language models. Following [4], we attack the estimation task in the continuous domain. The basic idea is to convert the word indices to a continuous representation and to use a probability estimator operating on this space. Since the resulting distributions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. Probability estimation and interpolation in a continuous space is mathematically well understood and many powerful algorithms are available that can perform meaningful interpolations even when only a limited amount of training material is available.

A first evaluation of such an approach using a neural network demonstrated that this technique could be incorporated into a conversational speech recognizer and reduce the word error rate [5]. However, the word error rate of the baseline system was rather high and one may wonder if this new language model can still achieve significant improvements once the system has been optimized to the task. This paper describes the integration of an improved neural LM in a baseline system that achieves state-of-the-art performance (a word error of under 25% on the SWB NIST Eval01 test set). The neural language model is used to rescore lattices after two acoustic model adaptation passes instead of being used to carry out a full decode as was reported previously. This improved neural language model is compared to a state-of-the-art backoff LM using modified Kneser-Ney smoothing.

The remainder of this paper is organized as follows. The next section describes the architecture of the continuous space language model. Section 3 summarizes the baseline speech recognizer and Section 4 explains how the neural LM is incorporated in the system. Comparative results are reported in Section 5 and the paper concludes with a discussion and some possible extensions of the approach.

2. CONNECTIONIST LM ARCHITECTURE

The architecture of the connectionist LM is shown in Figure 1. A standard fully-connected multi-layer perceptron is used. The inputs to the neural network are the indices of the $n-1$ previous words in the vocabulary $w_{j-n+1}, ..., w_{j-2}, w_{j-1}$ and the outputs are the posterior probabilities of *all* words of the vocabulary:

$$P(w_j = i | w_{j-n+1}, ..., w_{j-2}, w_{j-1}) \qquad \forall i \in [1, N]$$

where N is the size of the vocabulary. This can be contrasted to standard language modeling where each n-gram probability is calculated independently. The input uses the so-called 1-of-n coding, i.e., the *i-th* word of the vocabulary is coded by setting the *i-th* element of the vector to 1 and all the other elements to 0. This coding substantially simplifies the calculation of the projection layer since

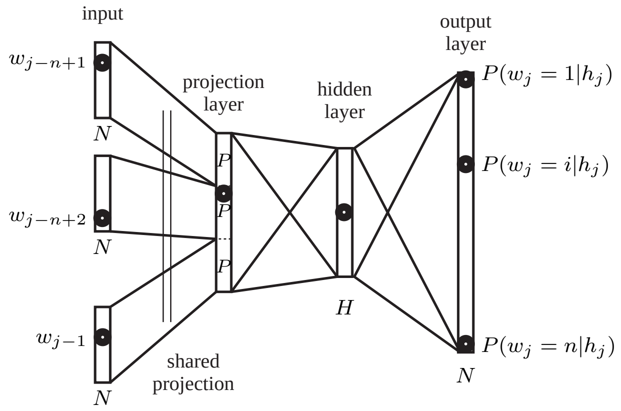

Fig. 1. Architecture of the connectionist language model. h_j denotes the context $w_{j-n+1}, ..., w_{j-1}$. P, H and N is the size of projection, hidden and output layer respectively. When shortlists are used the size of the output layer is much smaller then the size of the vocabulary.

we only need to copy the *i-th* line of the $N \times P$ dimensional projection matrix, where N is the size of the vocabulary and P the size of the projection. At the hidden layer a *tanh* non-linearity is used and the outputs are normalized using a softmax normalization to obtain posterior probabilities: the value of the *i-th* output neuron corresponds directly to the probability $P(w_j = i|h_j)$, where h_j denotes the word history $w_{j-n+1}, ..., w_{j-2}, w_{j-1}$. Training is performed with the back-propagation algorithm using cross-entropy as the error function. It can be shown that the neural network minimizes the perplexity on the training data (see [5] for more details of the architecture).

This neural LM has a rather high complexity when it is used to calculate the probability $P(w_j|h_j)$ of only one n-gram. The activities of the projection layer are obtained by a simple table look-up and can be neglected in the complexity analysis. The calculation of the hidden- and output-layer activities correspond to a matrix/vector multiplication followed by the application of a non-linear function. This gives the following number of floating point operations (Flops):

$$((n-1)P \times H) + H + (H \times N) + N$$

where H the size of the hidden layer. Since N is much larger than H, the complexity is dominated by the calculation at the output layer. For usual values of n=3, N=42k, P=50 and H=300, about 13 MFlops are needed to calculate one LM probability, which is computationally very expensive for full decoding. Note that due to the softmax normalization, all of the output activities need to be calculated even if only one probability is needed.

The proposed LM is used during the final decoding pass to rescore lattices. It is easy to arrange that all the different LM probabilities for the same context are requested sequentially. Using caching techniques, the neural LM can calculate these additional predictions for the same input context at no cost since they are already available at the output! In addition, many of the possible n-gram probabilities are never requested when rescoring lattices and it is not very reasonable to spend a lot of computation power on words that appear very rarely. Therefore, we chose to limit the output of the neural network to the 2000 most frequent words, referred to as a *shortlist* in the following discussion.

The LM probabilities of words in the shortlist are calculated by the network ($\hat{P}_N$) and the LM probabilities of the remaining words by a standard 4-gram backoff LM (P_B):

$$P(w_j|h_j) \quad = \begin{cases} \hat{P}_N(w_j|h_j) \cdot P_S(h_j) & \text{if } w_j \in \text{shortlist} \\ P_B(w_j|h_j) & \text{else} \end{cases}$$

$$\text{with} \quad P_S(h_j) = \sum_{w \in shortlist(h_j)} P_B(w|h_j)$$

In other words, one can say that the neural network redistributes the probability mass of all the words in the shortlist.[1] These probability masses can be precalculated and easily stored in the data structures of the standard 4-gram LM. A standard backoff technique is used if the probability mass for a requested input context is not directly available. Limiting the output of the neural network to the 2000 most frequent words, covers 75% of the requested 4-grams when calculating the perplexity of the Eval01 test corpus and about 85% when rescoring the lattices. Using these optimizations techniques lattices can be rescored in 1-2xRT, depending on their size.

3. BASELINE SYSTEM

The LIMSI conversational speech recognizer is derived from our broadcast news transcription system [3]. The word recognizer uses continuous density HMMs with Gaussian mixture for acoustic modeling and *n*-gram statistics estimated on large text corpora for language modeling. Each context-dependent phone model is a tied-state left-to-right CD-HMM with Gaussian mixture observation densities where the tied states are obtained by means of a decision tree. The acoustic feature vector has 39-components comprised of 12 cepstrum coefficients and the log energy, along with the first and second order derivatives. Cepstral mean and variance normalization are carried out on each conversation side. The acoustic models are trained on a total of about 230 hours of data from the LDC SWB1, CHE and SWB-CELL corpora.

3.1. Language model training

The baseline language model is constructed as follows: Separate backoff n-gram LMs were estimated on the following audio training corpora transcriptions: 2.7M words of the SWB1 LDC transcriptions, 2.9M words of SWB1 ISIP transcriptions, 230k words of SWB cellular training transcriptions, and 215k words of CallHome corpus transcriptions. Additional backoff LMs were built using 240M words of commercially produced BN transcripts and a subset of the BN training corpus similar in style to the SwitchBoard data (see [3] for details on the data selection method). The 4-gram backoff LMs were built using the modified version of Kneser-Ney smoothing as implemented in the SRI LM toolkit [6]. The LM vocabulary contains 41670 words.

A single backoff LM was built by merging these 6 models. The interpolation coefficients were estimated with an EM procedure. The resulting LM has 12M 2-grams, 26M 3-grams and 21M 4-grams. The perplexity on the Eval01 test data is 83.0 (the decompounded perplexity is 60.3). This interpolated model is our baseline SWB 4-gram LM. The bigram and trigram components of this LM are used in some of the early decoding passes.

[1] Note that the sum of the probabilities of the words in the shortlist for a given context is normalized $\sum_{w \in shortlist} \hat{P}_N(w|h_j) = 1$.

Corpus	*ISIP*	*LDC*	*CH*	*CELL*	*interpol. w BN*
backoff	115.7	113.7	189.2	151.2	83.0
neural	106.4	104.9	181.6	150.9	78.8

Table 1. Perplexities of the backoff and the neural 4-gram LM estimated on different transcription sets (SWB ISIP, SWB LDC, CallHome, and SWB Cellular).

The neural LM was only trained on the conversational speech corpora since its purpose is to do good interpolations when only little training data is available. Table 1 summarizes the perplexities of the different language models. The neural LM achieves relative perplexity reductions of up to 8% on all corpora. The perplexity of these LMs interpolated with the BN LMs is shown in the last column. The overall perplexity is 78.8 (58.0 decompounded).

3.2. Decoding

Decoding is carried out in 4 passes. In the first pass the speaker gender is identified for each conversation side using Gaussian mixture models, and a fast 3-gram decode is performed to generate approximate transcriptions. These transcriptions are only used to compute the VTLN warp factors for each conversation side. All of the following passes make use of the VTLN-warped data. Each subsequent decoding pass generates a 2-gram word lattice per speaker turn which is expanded with the 4-gram baseline backoff LM and converted into a confusion network with posterior probabilities. The best hypothesis in the confusion network is used in the next decoding pass for unsupervised MLLR adaptation [7] of the acoustic models. Two regression classes (speech and non speech) are used in the third pass, whereas 5 phonemic regression classes (non speech, voiceless consonants, voiced consonants, and two vowel classes) are used for the fourth pass. The baseline system has a word error rate of 25.0% on the NIST Eval01 test set (see pass 4 in Figure 2, top line).

In addition, two alternative sets of acoustic models were used in a fifth decoding pass (including MLLR adaptation with 5 regression classes): a model set based on MFCC features, and a model set based on PLP features but with short-term cesptral normalization (denoted PLP-S in Figure 2). The outputs corresponding to these 3 sets of acoustic models are finally combined using ROVER [8].

4. RESCORING LATTICES WITH THE NEURAL LM

In principle, the neural LM could completely replace the standard backoff LM and be used throughout all decoding passes and in our initial experiments a full decode with the neural LM was used [5]. To reduce the computational costs, an alternative solution is applied in this paper. First we generate lattices using the 2-gram backoff LM as described above. These lattice are then expanded with the baseline trigram and then pruned. Then the baseline backoff or the neural 4-gram LM is used to expand and rescore theses lattices. It is important to note that the neural LM never backs off to lower order n-grams since it can interpolate an LM probability for any possible context. Table 2 summarizes the resulting differences in lattice size. The neural LM generates lattices which are about 30% larger than the baseline LM. The use of a shortlist and caching techniques with careful optimization [5] makes it possible to rescore the lattices in less than 2xRT.

LM:	2-gram	3-gram	4-gram	
	backoff	backoff	backoff	neural
#nodes	528	476	709	1005
#links	2628	990	1932	3067

Table 2. Average per speaker turn lattice statistics on the NIST SWB Eval01 data. The expanded 3-gram lattices have been pruned.

	PLP	*MFCC*	*PLP-S*	*ROVER*
ML acoustic models:				
backoff LM	25.0%	24.9%	25.3%	24.2%
neural LM	24.5%	24.4%	25.0%	23.8%
MMI acoustic models:				
backoff LM	24.0%	24.1%	23.9%	23.3%
neural LM	23.6%	23.7%	23.7%	23.0%

Table 3. Word error rates on SWB Eval01 test data using either the baseline 4-gram backoff LM or the neural 4-gram LM.

Although the lattices have low oracle word error rate (below 5%), it is interesting to know what the word error rate would be if a perfect LM was available to rescore the lattices. If the number is close to the word error rate of the baseline system, it would not be worth pursuing research on better language models, but rather on improving the acoustic models. This perfect LM is not easy to get since it is not reasonable to build a LM directly on the 56k words of the Eval01 transcriptions. Therefore we first trained a neural LM on the full Switchboard corpus. After convergence, training was continued on the Eval01 data (using a small learning rate), doing by these means supervised LM adaptation.

The resulting LM has a perplexity of 22.2 on Eval01 data and lattice rescoring gives an word error rate of 16.6%. This means that there is room for advances in language modeling, but that significant improvements in acoustic modeling are also needed in order to reduce the word error rate to under 10%.

In the following section we first report results when using the neural LM to rescore only the final lattices after the five class acoustic model adaptation. On the other hand, one may argue that the best possible hypothesis should be used to perform unsupervised MLLR adaptation. Therefore it may be reasonable to already use the neural LM to rescore the lattices obtained by the 2nd and 3rd pass respectively in order to obtain better hypothesis for the adaptation passes.

5. RECOGNITION RESULTS

Table 3 summarizes the word error rates for the three different sets of acoustic models (with ML and MMI training) and their Rover combination when using the baseline backoff LM or the neural network LM based approach in the last decoding pass. Results are given on the NIST SWB Eval01 test set (6h of audio data) used as development data.

Although the neural language model only achieved a rather modest perplexity reduction (cf. Table 1), the word error rates decreased by 0.5% with the baseline ML PLP acoustic models (from 25.0% to 24.5%). This is in contrast to many reported works in the literature where new language models achieve large perplexity reductions, that unfortunately do not lead to word error improvements. We are also using MMI training in order to improve the

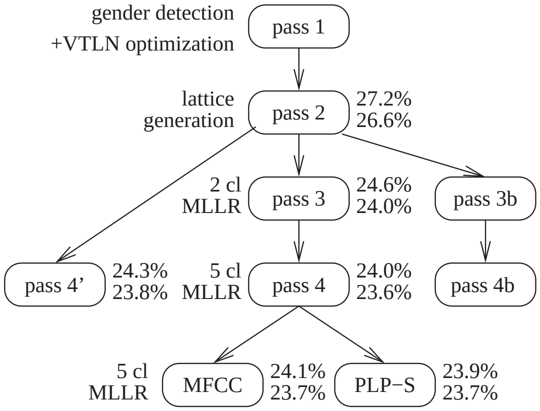

Fig. 2. Comparison of different decoding strategies (Eval01, MMI acoustic models). pass 3 & 4: MLLR using the hypothesis after rescoring with 4-gram backoff LM; pass 3b & 4b: MLLR using the hypothesis after rescoring with the neural LM. The numbers at the right side of each pass give the word error rates with the backoff and the neural LM respectively.

	PLP	*MFCC*	*PLP-S*	*ROVER*
backoff LM	27.3%	27.2%	26.9%	26.3%
neural LM	26.9%	26.9%	26.6%	25.9%

Table 4. Word error rates on eval02 for three MMI systems using different front-ends for the baseline and neural LMs.

acoustic models. Using the more accurate MMI acoustic models does not change the gain provided by using the neural language model as we obtain basically the same word error reduction of 0.4%. Note that a 0.4% word error reduction is not easy to obtain at this error level. This gain is basically the same as that achieved in the fourth decoding pass which requires ten times more computation (20xRT).

The improvement obtained by the neural LM is also maintained when the outputs for the three front-ends (PLP, MFCC, PLP-S) are combined with the ROVER algorithm [8] as shown in the last column of Table 3.

One can question if it is advantageous to use the neural LM to rescore the lattices of all passes or to apply it only in the last step. As can been seen in Figure 2 the neural LM also achieves comparable word error reductions when used to rescore the 2nd and 3rd pass. It turned also out that the final word error was identical (23.6%, following passes 3b and 4b) when the neural LM was used to rescore lattices in all passes instead of only in the last one. The backoff LM is seen to also have a lower word error in pass 4b since it takes advantage of the neural network rescoring in passes 2 and 3b.

We have also tried to perform only one MLLR adaptation pass using 5 regression classes (pass 4', Figure 2). This achieves a slightly higher word error rate of 23.8%, but without the 20xRT cost of the last adaptation pass.

Finally, Table 4 gives the results on the NIST SWB eval02 test set that was not used during development. The gain of the continuous space LM is very comparable to that observed on the development data.

6. DISCUSSION AND SUMMARY

We have described experiments with a neural language model that is well suited when only a small amount of LM training data is available. This makes it very interesting for conversational speech recognition. The approach seeks to achieve better estimation of the LM probabilities by performing the estimation in a continuous space, allowing by these means “smooth interpolations.”

The neural network language model has been extensively tested by rescoring the lattices of a conversational speech recognizer. Despite only small gains in perplexity with respect to a carefully tuned backoff LM, the neural LM achieved consistent word error improvements of over 0.4%. This word error reduction is maintained when the overall system is improved, e.g. unsupervised acoustic model adaptation, better acoustic modeling using MMIE or system combination.

The presented approach uses a neural network to project the words onto a continuous space and to estimate the LM probabilities. We are currently working on other probability estimators that operate on the continuous space. Promising candidates are for instance Gaussian mixture densities or RBF networks. Another interesting direction is to train an error corrective LM. This could be done by using a training criterion that seeks to minimize the word error after rescoring training data lattices.

7. ACKNOWLEDGMENTS

The authors would like to recognize the contributions of G. Adda, L. Chen, L. Lamel, F. Lefèvre, and D. Mas for their involvement in the development of the LIMSI conversational speech recogniton system on top of which this current work is based.

8. REFERENCES

[1] A. Lee, J. Fiscus, J. Garofolo, M. Przybocki, A. Martin, G. Sanders, and D. Pallett, “The 2002 NIST RT evaluation speech-to-text results,” in *Rich Transcription Workshop*, May 7 2002.

[2] R. Iyer and M. Ostendorf, “Relevance weighting for combining multi-domain data for n-gram language modeling,” *Computer Speech & Language*, vol. 13, pp. 267–282, 1999.

[3] J.L. Gauvain, L. Lamel, H. Schwenk, G. Adda, L. Chen, and F. Lefèvre, “Conversational telephone speech recognition,” ICASSP, 2003.

[4] Y. Bengio and R. Ducharme, “A neural probabilistic language model,” in *NIPS*. 2001, vol. 13, Morgan Kaufmann.

[5] H. Schwenk and J.L. Gauvain, “Connectionist language modeling for large vocabulary continuous speech recognition,” in *Proc. ICASSP*, 2002, pp. I: 765–768.

[6] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proc. ICSLP*, 2002, pp. II: 901–904.

[7] C.J. Leggetter and P.C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[8] J.G. Fiscus, “A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER),” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 347–354.