Building Continuous Space Language Models for Transcribing European Languages

Holger Schwenk and Jean-Luc Gauvain

LIMSI-CNRS
BP 133, bat 508, 91436 Orsay cedex, FRANCE
schwenk,gauvain@limsi.fr

10.21437/Interspeech.2005-14

Abstract

Large vocabulary continuous speech recognizers for English Broadcast News achieve today word error rates below 10%. An important factor for this succes is the availability of large amounts of acoustic and language modeling training data. In this paper the recognition of French Broadcast News and English and Spanish parliament speeches is addressed, tasks for which less resources are available. A neural network language model is applied that takes better advantage of the limited amount of training data. This approach performs the estimation of the probabilities in a continuous space, allowing by this means smooth interpolations. Word error reduction of up to 0.9% absolute are reported with respect to a carefully tuned backoff language model trained on the same data.

1. Introduction

Language models play an important role in many applications like character and speech recognition, translation and information retrieval. The dominant approach, at least in large vocabulary continuous speech recognition (LVCSR), are n-gram backoff language models (LM). These models are usual trained on huge amounts of data in state-of-the-art LVCSR for English Broadcast News (BN) and conversational telephone speech (CTS). For instance, most of the sites participating in the 2004 rich transcription evaluation used almost 2G words to build LM for English BN [1]. These large amounts of language model training data are usually not available for other major European languages like German, French, Italian or Spanish. This may be explained by the fact that less funding is available to collect the resources, but in some countries copyright issues also complicate the collection and transcription of large amounts of audio data or the use of text resources. The same data sparseness problems may arise when developing LVCSR for other tasks than BN. In this paper the problem of transcribing parliament speeches is addressed. Therefore LM techniques that perform well with a limited amount of training data are interesting for building LVCSR for these languages and tasks.

Usually, class language models are used when insufficient data is available. In this paper we describe the application of a new approach that uses a neural network to estimate the LM posterior probabilities [2, 3]. The basic idea is to project the word indices onto a continuous space and to use a probability estimator operating on this space. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and the n-gram probability estimation. This is still a n-gram approach, but the LM posterior probabilities are "interpolated" for any possible context of length n-1 instead of backing-off to shorter contexts. This approach has already been successfully used in a CTS and BN large vocabulary speech recognizer for the English language, achieving word error reductions of up to 0.5% absolute [4, 5, 6]. Here we will show that even better improvements can be obtained for the transcription of European parliament speeches in English and Spanish and the recognition of French Broadcast News. All the described system have been carefully tuned before the neural network LM is applied.

This work was partially financed by the European Commission under the FP6 Integrated Project TC-STAR.

2. Architecture

The architecture of the neural network n-gram 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 $h_j = w_{j-n+1}, ..., w_{j-2}, w_{j-1}$ and the outputs are the posterior probabilities of *all* words of the vocabulary:

$$P(w_j = i|h_j) \qquad \forall i \in [1, N] \tag{1}$$

where N is the size of the vocabulary. 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. The *i-th* line of the $N \times P$ dimensional projection matrix corresponds to the continuous representation of the i-th word.

Let us denote c_k these projections, d_j the hidden layer activities, o_i the outputs, p_i their softmax normalization, and m_{jk}, b_j, v_{ij} and k_i the hidden and output layer weights and the corresponding biases. Using matrix/vector notation the neural network performs the following operations:

$$d_j = \tanh\left(m_{jk}\, c_k + b_j\right) \tag{2}$$

$$o_i = \tanh\left(v_{ij}\, d_j + k_i\right) \tag{3}$$

$$p_i = e^{o_i} \;/\; \sum_{k=1}^{N} e^{o_k} \tag{4}$$

The value of the output neuron p_i corresponds directly to the probability $P(w_j = i|h_j)$. Training is performed with the standard back-propagation algorithm using cross-entropy as error function, and a weight decay regularization term. The targets are set to 1.0 for the next word in the training sentence and to 0.0 for all the other ones. It can be shown that the outputs of a neural network trained in this manner converge to the posterior

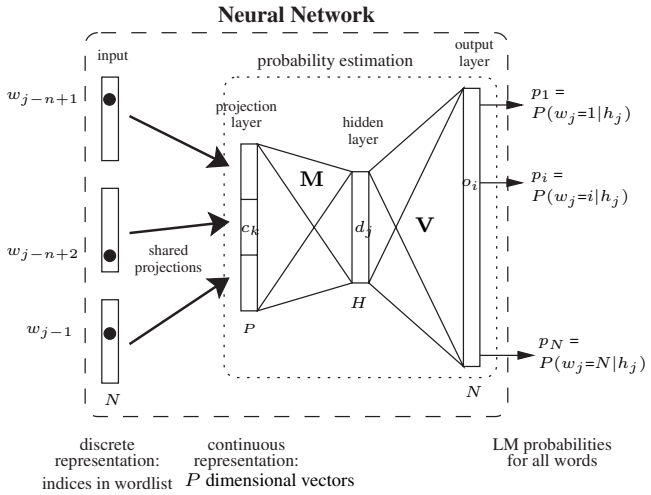

Figure 1: *Architecture of the neural network language model. h_j denotes the context $w_{j-n+1}, ..., w_{j-1}$. P is the size of one projection and H and N is the size of the hidden and output layer respectively. When shortlists are used the size of the output layer is much smaller then the size of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

probabilities. Therefore, the neural network directly minimizes the perplexity on the training data. Note also that the gradient is back-propagated through the projection-layer, which means that the neural network learns the projection of the words onto the continuous space that is best for the probability estimation task.

The complexity to calculate one probability with this basic version of the neural network LM is quite high, mainly due to the large output layer, and several improvements are necessary to make the model tractable for LVCSR [7]:

1. *Lattice rescoring*: decoding is done with a standard back-off LM and a lattice is generated. The neural network LM is then used to rescore the lattice.
2. *Shortlists*: the neural network is only used to predict the LM probabilities of a subset of the whole vocabulary.
3. *Regrouping*: all LM probability requests in one lattice are collected and sorted. By these means all LM probability requests with the same context h_t lead to only one forward pass through the neural network.
4. *Block mode*: several examples are propagated at once through the neural network, allowing the use of faster matrix/matrix operations.
5. *CPU optimization*: machine specific libraries BLAS are used for fast matrix and vector operations.

The idea behind shortlists is to use the neural network only to predict the s most frequent words, $s \ll |V|$, reducing by these means drastically the complexity. All words of the word list are still considered at the input of the neural network. The LM probabilities of words in the shortlist ($\hat{P}_N$) are calculated by the neural network and the LM probabilities of the remaining words ($\hat{P}_B$) are obtained from a standard 4-gram back-off LM:

$$\hat{P}(w_t|h_t) = \begin{cases} \hat{P}_N(w_t|h_t) \cdot P_S(h_t) & \text{if } w_t \in \text{shortlist} \\ \hat{P}_B(w_t|h_t) & \text{else} \end{cases} \quad (5)$$

$$P_S(h_t) = \sum_{w \in shortlist(h_t)} \hat{P}_B(w|h_t) \quad (6)$$

It can be considered that the neural network redistributes the probability mass of all the words in the shortlist. This probability mass is precalculated and stored in the data structures of the backoff LM. A back-off technique is used if the probability mass for a requested input context is not directly available.

During lattice rescoring LM probabilities with the same context h_t are often requested several times on potentially different nodes in the lattice. Collecting and regrouping all these calls prevents multiple forward passes since all LM predictions for the same context are immediately available at the output. Further improvements can be obtained by propagating several examples at once though the network, also known as bunch mode. In comparison to equation 2 and 3, this results in using matrix/matrix instead of matrix/vector operations which can be aggressively optimized on current CPU architectures.

Bunch mode has also been implemented for training of the neural network. Training of a typical network with a hidden layer with 500 nodes and a shortlist of length 2000 (about one million parameters) take less than one hour for one epoch through four million examples on a standard PC. When more training data is available, a hidden layer of more than thousand nodes is necessary since more capacity is needed. Usually we do not train one such big network, but several smaller ones and interpolate them together. More details can be found in [7].

3. Application to French Broadcast News

The neural network LM has already been used for CTS and BN recognition in English [5, 6]. Here we apply the same technique to a French BN system. The following resources have been used for language modeling:

- Transcriptions of the acoustic training data (4.0M words)
- Commercial transcriptions (88.5M words)
- Newspaper texts (508M words)
- WEB data (13.6M words)

First a LM was build for each corpus using modified Kneser-Ney smoothing as implemented in the SRI LM toolkit [8]. The individual LMs were then interpolated and merged together. An EM procedure was used to determine the coefficients that minimize the perplexity on the development data. Although the detailed transcriptions of the audio data represent only a small fraction of the available data, they get an interpolation coefficient of 0.43[1]. This shows clearly that the detailed audio transcriptions are the most appropriate text sources for the task and the neural network LM was first trained on the transcriptions only. In all cases, the neural network LM is interpolated with the reference backoff LM. In the following sections we first describe the use of the neural network LM in a 7xRT system, and show then that it can also be used in a fast 1xRT system.

3.1. Reference 7xRT system

The French BN system is based on techniques developed for the English BN system. The acoustic model uses tied-state position-dependent triphones trained on about 190 hours of BN data. Decoding is done in three passes including acoustic model adaptation (CMLLR and MLLR), pronunciation probabilities and consensus decoding (see [9] for more details). In our previous experiences on English BN and CTS a 65k vocabulary

[1]The other coefficients are 0.14 for the commercial transcripts, 0.35 for the newspaper texts and 0.08 for the WEB data.

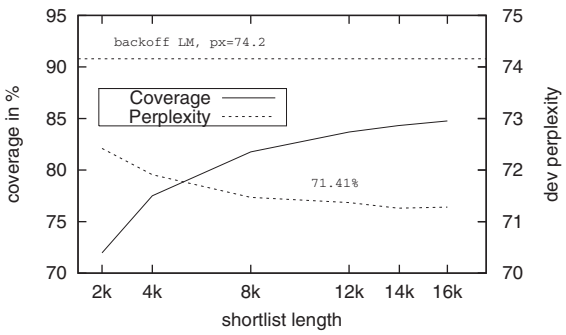

Figure 2: Coverage and perplexity on the dev data of the neural network LM in function of the size of the shortlist for French BN. The neural network LM is trained on 4M words only.

was used and the shortlist was of size 2000. This resulted in a coverage of about 90%, i.e. nine out of ten LM requests were processed by the neural network. The French BN system uses a word list of 200k and consequently larger shortlist sizes were investigated. Figure 2 shows the coverage and the perplexity of the neural network LM on the development data in function of the shortlist size. It can be clearly seen that a shortlist of length 2000 is insufficient for this task (the coverage is only 72%) and that better results can be obtained with larger output layers.

However, the curves flatten out with increasing size and a shortlist of length of 12k was used in most of the experiments. In this case the perplexity decreases from 74.2 to 71.4. Figure 3 shows the results when rescoring the lattices with the neural network LM. The coverage, i.e. the number of LM requests in the lattice done by the neural network LM, increases from 72% (length = 2k) to 88% (length = 16k). With 12k shortlists the word error decreases from 10.74% to 10.51% at an additional decoding time of 0.11xRT. Longer short lists do not lead to further word error reductions.

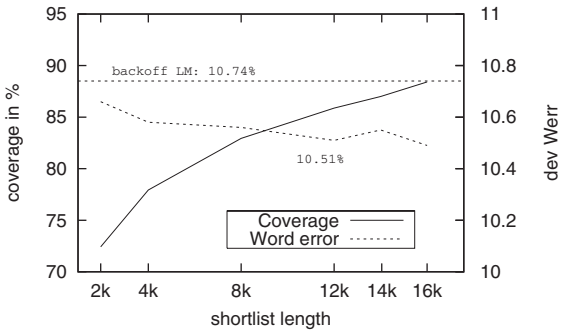

Figure 3: Coverage and word error on the dev data when rescoring lattices with the neural network LM in function of the size of the shortlist for French BN. The neural network LM is trained on 4M words only.

In a second set of experiments the neural network LM was trained on more data by adding about 16M words of commercial transcriptions, giving a total of 21M words. In this case four neural networks were trained on the data and then interpolated together. This results in faster training an gives usually better results than training one neural network with a large hidden layer [7]. Table 1 summarizes the performances of the different

	Backoff LM	Neural LM	
LM data	614M	4M	21M
Perplexity dev	74.2	71.4	71.2
Word error dev	10.74%	10.51%	10.40%
eval	12.45%	12.18%	12.03%
addtl. runtime	-	0.11xRT	0.45xRT

Table 1: Results for the French 7xRT BN system.

language models. Although there is only a small change in perplexity when the neural network LM trained on more data, an additional word error reduction of 0.11% is achieved.

It is also interesting to note the good generalization behavior of the neural network LM. Network selection and tuning of the parameters have been done on the development data, but the word error reduction obtained on the evaluation data (-0.42%), that was never used during developement, is even better than the one obtained on the development data itself (-0.34%).

3.2. Fast 1xRT system

Building a real time Broadcast News system is a challenging task since the time constraint makes it very difficult to use all the techniques that help to get low word error rates. It is for instance difficult to do several decoding passes with acoustic model adaptation. The system used here uses only one decoding pass (see [9] for more details). The neural network LM for this system was trained on the same data than the 7xRT system. Due to the real time requirements a short list size of 8k words was used (the complexity of the neural network LM increases linearly with the shortlist length) and the hidden layer was of size 500. This resulted in an additional time of 0.05xRT to rescore the lattices. The coverage when rescoring lattices with the neural network LM is 85.2%. Table 2 summarizes the results of this system.

	backoff LM	Neural LM	
LM data	614M	4M	21M
Perplexity dev	70.2	67.7	68.3
Word error dev	14.24%	14.02%	13.88%
eval	17.08%	16.85%	16.78%
addtl. runtime	-	0.05xRT	0.05xRT

Table 2: Results for the French 1xRT BN system.

Despite the smaller networks the same word error reduction than with the 7xRT French BN system was observed.

4. Application to Parliament Speeches

The European project TC-STAR is concerned with speech to speech translation of European parliament speeches. The main focus is on the translation directions European English to Spanish and vice versa. In this paper we describe the efforts undertaken to build language models for the speech recognition systems for these languages. In both cases, the main source for language model training are the transcripts of the acoustic training data (about 350k words) and the official translations of parliament speeches as distributed by the European Community (about 33M words per language). These parallel texts are by the way also used for the statistical translation engines.

4.1. European English system

The speech recognizer for the English parliament speeches has a similar architecture than the French BN system. The incorporation of the neural network LM was again done by rescoring the final lattices. The 4-gram backoff LM and the neural network LM were trained on in-domain data only: 40h of transcribed audio data and 32M words of English parliament speeches for language model training. In these experiments, the neural network was trained on the same amount of data than the backoff LM, using a shortlist of 2000 words. The coverage is 81.4% on the development data and 83.7% when rescoring lattices. The vocabulary has 42k words, resulting in an OOV rate on the development data of 0.41%. The results are summarized in table 3.

	Backoff LM	Neural LM
LM data	32M	32M
Perpl dev	99.7	87.8
Werr dev	12.13%	11.26%
eval	12.04%	11.04%
addtl. time	-	0.08xRT

Table 3: Recognition of English parliament speeches.

A perplexity reduction of 12% relative was obtained (99.7 $\rightarrow$ 87.8) and the word error rate improved by as much as 0.87% absolute (12.13 $\rightarrow$ 11.26%). The additional processing time needed to rescore the lattices is less than 0.1xRT. Again, the neural network LM shows a very good generalization behavior: with 1.0% absoulte the word error reduction obtained on the evaluation data is higher than the one on the development data.

4.2. Spanish system

The speech recognizer for the Spanish parliament speeches has the same structure than the English system. The only data used for the language model are the transcriptions of the audio data and the translated parliament speeches (33.5M words in total). A 64k vocabulary was used and the OOV rate on the development data is 0.60%. A shortlist of 2000 words was used, giving a coverage of 82.0% on the development data and 80.0% when rescoring lattices. Table 4 summarizes the results. The neural network LM achieves an improvement in perplexity of 10% relative and a word error reduction of 0.59% absolute. This gain is smaller than with the neural network LM for the English system. This may be explained by the smaller coverage during lattice rescoring and by the fact that the lattices themselves are smaller: for Spanish they have 307 arcs and 550 links in average, while there are 357 arcs and 698 links for the English system.

	Backoff LM	Neural LM
LM data	33.5M	33.5M
Perplexity dev	81.0	71.8
Word error dev	10.64%	10.05%
eval	11.55%	11.07%
addtl. runtime	-	0.07xRT

Table 4: Results for Spanish parliament speeches.

5. Conclusion

This paper addressed language modeling for large vocabulary speech recognition with limited amounts of in-domain language modeling data. The main idea is to perform the estimation of the LM probabilities in a continuous space, allowing by these means "smooth interpolations" in order take better advantage of the available training data. A neural network is used to learn simultaneously these projections and the probability estimation. Recognition is done by rescoring lattices after the last decoding pass which takes usually less than 0.1xRT.

The approach has been applied to three tasks: the recognition of French Broadcast News and the transcriptions of English and French speeches of the European Parliament. All system run in less than 10xRT and achieve word error rates in the range 10-12% with a backoff LM. This could be reduced by up to 0.9% absolute using the neural network LM. The approach has also been applied to a fast 1xRT French BN system, achieving word error reductions of 0.4% at an additional decoding cost of 0.05xRT. In all cases the neural network LM showed good generalization behavior: the word error reduction achieved on the evaluation data was often higher than the one obtained on the development data.

6. Acknowledgment

The authors would like to recognize the contributions of G. Adda, M. Adda, E. Bilinski, O. Galibert and L. Lamel for their involvement in the development of the speech recognition systems on top of which this work is based.

7. References

[1] J. Fiscus, J. Garofolo, A. Lee, A. Martin, D. Pallett, M. Przybocki, and G. Sanders, "Results of the fall 2004 STT and MDE evaluation," in *DARPA Rich Transcription Workshop, Palisades NY*, Nov 2004.

[2] Y. Bengio and R. Ducharme, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems*, vol. 13. Morgan Kaufmann, 2001.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1137–1155, 2003.

[4] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2002, pp. I: 765–768.

[5] ——, "Neural network language models for conversational speech recognition," in *International Conference on Speech and Language Processing*, 2004, pp. 1215–1218.

[6] ——, "Using neural network language models for lvcsr," in *2004 Rich Transcriptions Workshop, Pallisades, NY*, 2004.

[7] H. Schwenk, "Efficient training of large neural networks for language modeling," in *IEEE joint conference on neural networks*, 2004, pp. 3059–3062.

[8] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *International Conference on Speech and Language Processing*, 2002, pp. II: 901–904.

[9] J. Gauvain, G. Adda, M. Adda-Decker, A. Allauzen, V. Gendner, L. Lamel, and H. Schwenk, "Where are we in transcribing bn french?" submitted to Eurospeech 2005.