Large Vocabulary SOUL Neural Network Language Models

Hai-Son Le[1,2]*, Ilya Oparin*[2]*, Abdel Messaoudi*[2]*, Alexandre Allauzen*[1,2]*,*
Jean-Luc Gauvain[2]*, François Yvon*[1,2]

[1]Université Paris-Sud
[2]LIMSI CNRS, Spoken Language Processing Group
B.P. 133, 91403 Orsay, cedex, France
{lehaison,oparin,abdel,allauzen,gauvain,yvon}@limsi.fr

10.21437/Interspeech.2011-259

Abstract

This paper presents continuation of research on Structured OUtput Layer Neural Network language models (SOUL NNLM) for automatic speech recognition. As SOUL NNLMs allow estimating probabilities for all in-vocabulary words and not only for those pertaining to a limited shortlist, we investigate its performance on a large-vocabulary task. Significant improvements both in perplexity and word error rate over conventional shortlist-based NNLMs are shown on a challenging Arabic GALE task characterized by a recognition vocabulary of about 300k entries. A new training scheme is proposed for SOUL NNLMs that is based on separate training of the out-of-shortlist part of the output layer. It enables using more data at each iteration of a neural network without any considerable slow-down in training and brings additional improvements in speech recognition performance.

Index Terms: Neural Network Language Model, Automatic Speech Recognition, Speech-To-Text

1. Introduction

Conventional n-gram language models (LMs) are the basis of modern language modeling for speech-to-text (STT). There are quite few approaches that were shown to systematically bring additional improvements over an n-gram baseline and are thus used in large-scale STT systems.

One of the most successful approaches to date is to model and use distributed word representations on top of conventional n-gram models. Contrary to n-grams that rely on a discrete space representation of the vocabulary, where each word is associated with a discrete index, distributionally similar words can be represented as neighbors in a continuous space. This turns n-grams distributions into smooth functions of word representations. These representations and the associated probability estimates are jointly computed in a Neural Network Language Model (NNLM). The use of neural-networks for language modeling was introduced in [1] and successfully applied to speech recognition [2, 3, 4].

The major difficulty with the neural network approach remains the complexity of inference and training, which largely depends on the size of the output vocabulary (i.e. words that can be predicted). One practical solution is to restrict the output vocabulary to a shortlist composed of the most frequent words (usually from 8k up to 20k). Probabilities of all n-grams finished with out-of-shortlist words are estimated with a baseline n-gram model [3]. Such a restriction is likely to limit the potential of NNLMs. To circumvent this problem we have recently proposed Structured Output Layer (SOUL) NNLMs that were shown to improve over state-of-the-art shortlist NNLMs [5]. The SOUL model combines neural network approach with another successful language modeling approach, namely class-based LMs (as introduced in, e.g. [6]). Structuring the output layer of the model and using word class information makes estimating all in-vocabulary words with a NNLM computationally feasible.

Our previous results were reported on the GALE Mandarin task with a 56k word recognition vocabulary. As estimating all in-vocabulary word probabilities is probably the major distinctive feature of SOUL NNLMs, it seems interesting to evaluate the performance on a larger vocabulary task. The intuition behind is that the relative coverage of shortlist-based NNLMs might be smaller, leaving more space for the improvement with SOUL NNLMs. The GALE Arabic task is thus chosen to carry out STT experiments and evaluate the performance. Well-tuned LIMSI Arabic STT system is used in the experiments presented in this paper. It is based on a 4-gram Kneser-Ney discounted LM trained on about 2 billion corpora (without any pruning and cut-offs) interpolated with standard 12k shortlist NNLMs. The recognition vocabulary contains about 300k MADA decomposed entries.

A new training scheme for the class part of SOUL NNLMs is also presented in this paper. It makes a better use of available data during training and is aimed to provide more robust parameter estimates for large vocabulary tasks. Application of this scheme results in additional improvements in perplexity and speech recognition performance.

This paper is organized as follows. Previous and related work on SOUL and hierarchical NNLMs is briefly summarized in Section 2. The new training scheme for SOUL NNLMs is introduced in Section 3. Experiments and results are presented in Section 4. Section 5 provides a discussion and conclusions.

2. Previous and Related Work

A hierarchical structure of the neural network output layer was introduced in [7] (and further investigated in [8]) in order to speed-up training and inference. In this approach, the output vocabulary is first clustered and represented by a binary tree. Each internal node of the tree holds a word cluster which is binary divided in sub-clusters. Leaves correspond to words at the end of this recursive representation of the vocabulary. Thus the neural network aims to estimate probabilities of the paths in this binary tree given the history, rather than words directly.

We have recently proposed a new SOUL approach to structuring the output layer that avoids the constraint of a binary tree structure [5].

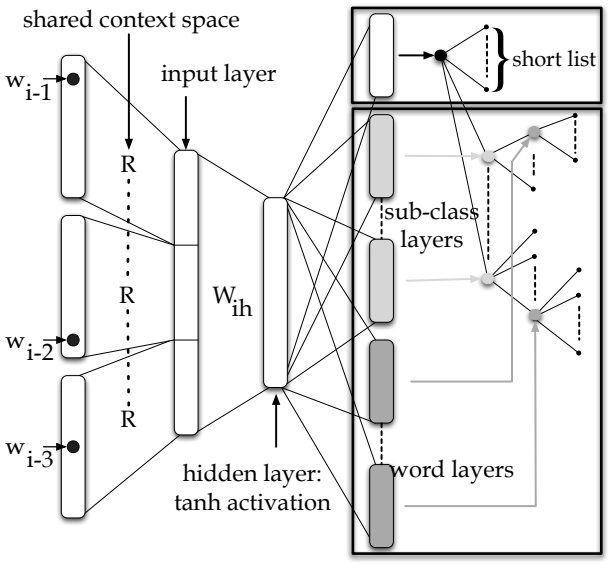

Figure 1: *The architecture of a SOUL Neural Network language model.*

The output vocabulary is structured by a clustering tree, where each word belongs to only one class and its associated sub-classes. If w_i denotes the i^{th} word in a sentence, the sequence $c_{1:D}(w_i) = c_1, \ldots, c_D$ encodes the path for the word w_i in the clustering tree, with D being the depth of the tree, $c_d(w_i)$ a class or sub-class assigned to w_i, and $c_D(w_i)$ being the leaf associated with w_i (the word itself). Then the n-gram probability of w_i given its history h can be estimated as follows:

$$P(w_i|h) = P(c_1(w_i)|h) \prod_{d=2}^{D} P(c_d(w_i)|h, c_{1:d-1}) \quad (1)$$

This equation holds as each words belongs to only one class, there is a softmax function at each level of hierarchical representation and each word ends up forming its own class in some leaf of the tree.

The overall architecture of a SOUL NNLM is presented in Figure 1. The model can be roughly divided into two parts. The shortlist part deals with in-shortlist words that form separate classes on their own without any sub-clustering. The softmax function in this first output layer spans over all shortlist words plus one additional node. This node serves as a root for a tree that deal with with all other words from the vocabulary and includes multiple sub-class layers with a softmax in each.

The SOUL NNLM approach was previously evaluated on GALE Mandarin STT task with a well-tuned baseline LIMSI STT system (as described in [9]) based on 56k recognition vocabulary. The gains attained with SOUL NNLMs correspond to a relative improvement of 23% in perplexity and 7-9% in CER over the baseline 4-gram Kneser-Ney discounted model when SOUL NNLMs are interpolated with the latter. As compared to conventional shortlist NNLMs, 0.1% absolute CER improvement for 4-grams and 0.2% for 6-grams was attained in cases the well tuned baseline has a character error rate under 10% [5].

Simultaneously with the work described above, using class information at the output layer was proposed for recurrent NNLMs [10]. This serves as an additional indication of continuous interest in improving neural network approach for STT.

3. SOUL NNLM Training Issues

Resampling of training data is conventionally used as it is computationally infeasible to train a NNLM on the same amounts of data as a baseline n-gram LM. Usually data up to 30M words are chosen after resampling at each iteration of a neural network. We are not aware of STT neural network language modeling experiments with large vocabularies that made use of significantly larger amounts of data, due to the prohibitive increase in computational load and training time.

As one deals with large vocabularies (e.g. as the one used in this study), the number of parameters related to the class-part of the model increases significantly. In this situation training data after resampling that is used for neural network training at each epoch may be insufficient to obtain robust parameter estimates.

We can consider the output layer of SOUL model consisting of two parts: the main softmax layer which models the probabilities of most frequent in-shortlist words (corresponds to the upper rectangle in the Figure 1) and the remaining softmax layers that deal with the classes of less frequent out-of-shortlist words (represented in the lower rectangle in the Figure 1). These two parts are not equally learned. For each example, while all parameters of main softmax layer are updated, only a little part of the other parameters are learned. The rate of update depends directly on word frequency. That means only the parameters related to shortlist words are frequently learned and this is not the case for the others. At the same time the number of parameters of the second part are much larger. As a result, following the previous training scheme there is one part of the output layer that is well learned and another part which is not. Thus we propose a new enhanced SOUL NNLM training scheme based on separate training of the out-of-shortlist part at the output layer. As compared to the standard SOUL NNLM training procedure, it includes one additional step.

Training of SOUL NNLM training can be summarized in four steps:

1. Shortlist NNLM training.
2. Dimensionality reduction.
3. Clustering based on distributed representations induced from the context layer of the neural network.
4. Whole vocabulary SOUL NNLM training.

In our experiments, the algorithm starts with 4k classes in addition to the shortlist of 8k recognition units. This makes 12k units in total, that is equal to the shortlist size for Arabic shortlist NNLMs (see results in Section 4). A standard NNLM model with the shortlist as an output is trained at step 1, following one vector initialization scheme [11]. Standard back-propagation training as in [12] is performed for the whole vocabulary SOUL NNLM at step 4.

The enhanced training scheme can be introduced as follows. After the clustering step 3, a new step $3'$ is added. This step is similar to 1 but is carried out only for out-of-shortlist words. The part associated with shortlist words is temporally kept intact. It reduces the size of first softmax layer to number of main classes of less frequent words (4k, 2 times smaller than 8k, the total size of the output layer at step 1). That is important since training time mostly depends on the size of the first softmax layer. Furthermore, the example number of out-of-shortlist words in training data is approximately 5 times smaller. It means that if we keep the training time for each epoch of this step as in step 1, we can increase the number of examples for each epoch by the factor of 10 without considerable slow-downs in training time.

4. Experiments and Results

Arabic GALE task characterized by a vocabulary of about 300k entries was chosen to evaluate the SOUL NNLM performance for large vocabularies. State-of-the-art LIMSI Arabic STT system is used to perform speech recognition experiments.

Arabic is a highly inflective and morphologically rich language. In order to deal with its peculiarities, decomposition of words in its morphological constituents was shown to improve speech recognition results [13, 14]. There are several approaches to Arabic word decomposition. In this study we use one of the most popular tools up-to-date, namely MADA: Morphological Analysis and Disambiguation for Arabic (http://www1.ccls.columbia.edu/~cadim/MADA.html).

Acoustic models are based on concatenated TRAP-DCT (as introduced in [15]), PLP and F0 features and are discriminatively trained on about 2000 hours of speech.

Language model training data consists of about 1.7 billion words before decomposition. Altogether 32 interpolated Kneser-Ney 4-gram LMs for different text corpora are trained on MADA-decomposed units (about 2 billion on total) without pruning and cut-offs. These models are further interpolated into the final LM that serves as a robust baseline model. Lattices generated with this model are subsequently rescored with NNLMs.

Three NNLMs are trained with different resampling parameters, sizes of context (200, 300, 400) and hidden layers (500, 400, 300) for each n-gram order. These models are interpolated in order to obtain final NNLMs. Resampling favors corpora containing broadcast news (*bn*) and broadcast conversations (*bc*) data as target data. Up to 30M words data are used at each NNLM iteration of shortlist-based and standard SOUL NNLMs. For enhanced SOUL NNLMs up to 300M words are used at the step $3'$ (see Section 3) to train class output layers that deal with out-of-shortlist words.

All training parameters for the SOUL NNLMs were kept the same as for the shortlist NNLMs. Thus it can be argued that the difference in performance is due to the use of the whole vocabulary at the output layer. Three GALE development and evaluation sets are used to evaluate the performance of different models, namely *dev09s*, *eval10ns* and *dev10c*. These sets consist of 23576, 45629 and 52181 MADA decomposed units respectively. Recognition vocabulary contains 296772 entries.

Perplexity and word error rate (WER) results are presented in Tables 1 and 2 respectively. The first rows in the tables correspond to the 4-gram baseline Kneser-Ney LM. For the shortlist-based NNLMs (marked with *shrtlst*) 12k most frequent words form the shortlist. Six-gram NNLMs (*6-gr*) were trained in order to verify possible improvements from using longer context as opposed to the usual 4-gram (*4-gr*) setup. Perplexity results in Table 1 are given both for stand-alone NNLMs (columns *s/a*) and for the cases NNLMs are interpolated with the baseline 4-gram LM (columns *int*). Models marked as *SOUL* are conventional SOUL NNLMs as introduced in [5] while *SOUL+* corresponds to the SOUL NNLMs that make use of the enhanced training scheme as described in Section 3. The latter use more data to train the out-of-shortlist part of output layers.

Perplexity results in Table 1 show that using longer context brings improvements both for shortlist and SOUL NNLMs. It should be mentioned that according to our experience longer-context conventional n-gram models bring only marginal improvements and at the same time result in a drastic increase in model sizes that makes them hard to handle. The ability of neural network LMs to improve performance with the increase of context goes in line with results obtained with recurrent NNLMs [4]. The latter can be regarded as a special case of neural networks that takes account of all the history seen before the predicted word.

Table 1: *Perplexity for different language models.*

LM type	dev09s		eval10ns		dev10c	
	s/a	int	s/a	int	s/a	int
4-gram baseline	312		239		256	
shrtlst NN 4-gr	324	276	247	213	256	224
SOUL NN 4-gr	293	256	225	200	231	208
SOUL+ NN 4-gr	277	250	214	195	221	204
shrtlst NN 6-gr	302	263	228	202	236	210
SOUL NN 6-gr	255	231	196	180	200	186
SOUL+ NN 6-gr	245	227	189	177	194	183

Perplexity also shows that SOUL NNLMs consistently outperform shortlist NNLMs of the same orders on all the test sets. Relative improvements about 10% for stand-alone NNLMs and 7% for interpolated models are observed for 4-gram NNLMs on different test sets. For longer-context 6-gram NNLMs, the gains from using SOUL NNLMs are even larger, about 15% and 12% in stand-alone and interpolated scenarios respectively.

It is also worth noticing that SOUL NNLMs (both 4-gram and 6-gram) outperform in terms of perplexity the baseline 4-gram LM trained on much bigger data.

Only minor gains in perplexity are attained with the enhanced SOUL NNLM training scheme as compared to standard SOUL NNLMs. This points out that using 10 times more data to train out-of-shortlist part of SOUL NNLMs does not have much influence on model performance.

Table 2: *WER for different language models (in %).*

LM type	dev09s	eval10ns	dev10c
4-gram baseline	14.8	9.6	14.5
shrtlst NN 4-gr	14.4	9.1	14.2
SOUL NN 4-gr	14.3	9.0	14.0
SOUL+ NN 4-gr	14.1	9.1	14.0
shrtlst NN 6-gr	14.3	9.1	14.2
SOUL NN 6-gr	14.0	8.9	14.0
SOUL+ NN 6-gr	14.0	8.9	13.9

Results in Table 2 show that the improvements in perplexity attained with SOUL NNLMs over shortlist NNLMs carry over to speech recognition experiments. The lattices generated with the baseline 4-gram Kneser-Ney LM were rescored with NNLMs of different types. The interpolation weights were tuned on GALE Phase 5 development data and are given in Table 3. It shows that the SOUL NNLMs obtain higher interpolation weights as compared to the shortlist NNLMs.

As can be seen from Table 2, using SOUL NNLMs results in better recognition performance as compared to shortlist NNLMs both for 4-gram and 6-gram cases. The gains from using 6-gram NNLMs are smaller than it could be expected as the lattices had to be pruned before rescoring with 6-grams due to computational reasons. The effect of pruning is most notable on *dev10c* set. This set contains some large lattices that need more severe pruning. However, as 6-gram shortlist NNLMs show no improvement with pruned lattices over 4-gram NNLMs, 6-gram SOUL NNLMs do still improve the results.

Enhanced SOUL NNLMs bring improvements over standard SOUL NNLMs on some datasets and NNLM configurations (4-gram NNLM on *dev09s* and 6-gram NNLM on *dev10c*). There is rather unexpected small degradation with 4-grams on *eval10ns* that is difficult to explain and we are going to run additional experiments to find out the reasons.

Table 3: *Neural network language model weights for interpolation with the baseline n-gram model.*

NNLM type	interpolation weight
shrtlst NN 4-gr	0.50
SOUL NN 4-gr	0.68
SOUL+ NN 4-gr	0.72
shrtlst NN 6-gr	0.55
SOUL NN 6-gr	0.74
SOUL+ NN 6-gr	0.75

5. Conclusions and Discussion

In this paper we have investigated performance of structured output layer neural networks on Arabic GALE task characterized by a large vocabulary containing approximatively 300k entries. In our previous work significant gains with SOUL NNLMs were reported on Mandarin GALE task with a smaller vocabulary of 56k words. As for Mandarin, the results on Arabic show that SOUL NNLMs consistently outperform conventional shortlist NNLMs both in terms of perplexity (up to 15% relative improvement) and recognition error rate (up to 0.3% absolute improvement) on a challenging GALE task.

As the Arabic vocabulary is 6 times larger than the Mandarin one, larger gains could be expected from using SOUL NNLMs, as they estimate probabilities for all n-grams and not only for those ending with a in-shortlist word. However, similar gains were observed.

One of the possible reasons for similar gains on Arabic as compared to Mandarin Chinese could be insufficient data to robustly estimate parameters in out-of-shortlist words in SOUL NNLMs as the number of these parameters is greatly increased with a much larger vocabulary. Thus enhanced SOUL NNLM training scheme was proposed in this paper. It supposes separate training of the part related to in-shortlist words (as each of these forms a separate class itself without sub-clustering) and the class part that deals with all other words from the vocabulary. One order of magnitude more data is used to train the out-of-shortlist part of SOUL NNLMs without drastic increase in computational charge and training time. However, only minor improvements in perplexity and WER were observed.

As increasing training data for more robust estimates of out-of-shortlist output layers of a SOUL NNLM doesn't seem to bring consistent significant improvements in recognition accuracy, another reason may be suggested. It is concerned with a relatively hight data coverage with shortlists. Shortlists are formed on the basis of all training data before resampling. For 12k shortlists used in Mandarin (as introduced in [5]) and Arabic NNLMs (described in this paper), coverage on the basis of all training data is 95% and 90% respectively. The overall coverage shows that though Arabic vocabulary is several times larger than the Chinese one, the shortlist coverage is only 5% lower. As training data is resampled at each NNLM epoch with the emphasis on sources containing target *bn* and *bc* data, the number of calls to a NNLM (i.e. coverage at each epoch) changes. The same is valid for development data due to the fact it consists only of *bn* and *bc* data and the general vocabulary may have different coverage. We checked the coverage on development and test data and observed no significant difference as compared to the overall coverage.

The shortlist coverage statistics shows that similar size shortlists do relatively well in terms of data coverage even for models with much larger vocabularies. Thus, as confirmed by the experimental results, improvements from using full-vocabulary SOUL NNLMs, while being consistent, are not proportional to a vocabulary size.

6. Acknowledgments

This work has been partially supported by OSEO under the Quaero program and by the GALE program. Any opinions, findings or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

7. References

[1] Bengio, Y., Ducharme, R. and Vincent, P., "A Neural Probabilistic Language Model", in Neural Information Processing Systems, 13:933-938, 2001.

[2] Schwenk, H. and Gauvain, J-L., "Connectionist Language Modeling for Large Vocabulary Continuous Speech Recognition", in Proc. of ICASSP'02, 765-768, 2002.

[3] Schwenk, H., "Continuous Space Language Models", in Computer, Speech & Language, 3(21):492-518, 2007.

[4] Mikolov, T., Karafiát, M., Burget, L., Černocký, J. and Khudanpur, S., "Recurrent Neural Network Based Language Model", in Proc. of Interspeech'10, 1045-1048, 2010.

[5] Le, H.-S., Oparin, I., Allauzen, A., Gauvain, J.-L. and Yvon, F., "Structured Output Layer Neural Network Language Model", in Proc. of ICASSP'11, 5524-5527, 2011.

[6] Brown, P.F., de Souza, P.V., Mercer, R.L., Della Pietra, V.J. and Lai, J.C., "Class-based n-gram Models of Natural Language", in Computational Linguistics, 4(18):467-479, 1992.

[7] Morin, F. and Bengio, Y., "Hierarchical Probabilistic Neural Network Language Model", in Proc. of AISTATS'05, 246-252, 2005.

[8] Mnih, A. and Hinton, G.E., "A Scalable Hierarchical Distributed Language Model", in Neural Information Processing Systems, 21:1081-1088, 2008.

[9] Lamel, L., Gauvain, J-L., Bac Le, V., Oparin, I. and Meng, S., "Improved Models for Mandarin Speech-to-Text Transcription", in Proc. of ICASSP'11, 4660-4663, 2011.

[10] Mikolov, T., Kombrink, S., Burget, L., Černocký, J. and Khudanpur, S., "Extensions of Recurrent Neural Network Language Model", in Proc. of ICASSP'11, 5528-5531, 2011.

[11] Le, H.S., Allauzen, A., Wisniewski, G. and Yvon, F., "Training Continuous Space Language Models: Some Practical Issues", in Proc. of EMNLP'10, 778-788, 2010.

[12] Bengio, Y., Ducharme, R., Vincent, P. and Janvin, Ch., "A Neural Probabilistic Language Model", in JMLR, 3:2003, 1532-4435, 2003.

[13] Lamel, L., Messaoudi, A. and Gauvain, J-L., "Investigating Morphological Decomposition for Transcription of Arabic Broadcast News and Broadcast Conversation Data", in Proc. of Interspeech'08, 1429-1432, 2008.

[14] Diehl, F., Gales, M.J.F., Tomalin, M. and Woodland, P.C., "Morphological Analysis and Decomposition for Arabic Speech-to-Text Systems", in Proc. of Interspeech'09, 2675-2678, 2009.

[15] Schwarz, P., Matějka, P. and Černocký, J., "Towards Lower Error Rates in Phoneme Recognition", in Proc. of TSD'04, Lecture Notes in Computer Science, 3206:465-472, 2004.