INVESTIGATION OF BACK-OFF BASED INTERPOLATION BETWEEN RECURRENT NEURAL NETWORK AND N-GRAM LANGUAGE MODELS

X. Chen, X. Liu, M. J. F. Gales, and P. C. Woodland

University of Cambridge Engineering Department, Cambridge, U.K.
{*xc257,xl207, mjfg, pcw*}*@eng.cam.ac.uk*

ABSTRACT

Recurrent neural network language models (RNNLMs) have become an increasingly popular choice for speech and language processing tasks including automatic speech recognition (ASR). As the generalization patterns of RNNLMs and n-gram LMs are inherently different, RNNLMs are usually combined with n-gram LMs via a fixed weighting based linear interpolation in state-of-the-art ASR systems. However, previous work doesn't fully exploit the difference of modelling power of the RNNLMs and n-gram LMs as n-gram level changes. In order to fully exploit the detailed n-gram level complementary attributes between the two LMs, a back-off based compact representation of n-gram dependent interpolation weights is proposed in this paper. This approach allows weight parameters to be robustly estimated on limited data. Experimental results are reported on the three tasks with varying amounts of training data. Small and consistent improvements in both perplexity and WER were obtained using the proposed interpolation approach over the baseline fixed weighting based linear interpolation.

Index Terms— Speech recognition, perplexity, language model interpolation, recurrent neural network

1. INTRODUCTION

Statistical language models (LMs) are vital components of speech and language processing systems designed for tasks such as speech recognition and machine translation. Back-off n-gram LMs are most widely used form of language models during last several decades due to their simple model structures, efficient parameter estimation and discounting techniques. When large quantities of training data are available, good generalization performance can be obtained using back-off n-gram LMs. A key part of the statistical language modelling problem for many tasks including speech recognition is to appropriately model the long-distance context dependencies. This usually presents a severe data sparsity problem for n-gram LMs. In order to address this issue, language modelling techniques that can represent longer span preceding history contexts in a continuous and lower dimensional vector space, for example, recurrent neural network language models (RNNLMs) can be used. RNNLMs have been reported to yield consistent performance improvements over back-off n-gram LMs across a range of tasks [1, 2, 3, 4, 5, 6, 7].

The intrinsic modelling characteristics and generalisation patterns of n-gram LMs and RNNLMs are different and complementary to each other. In order to draw strengths from both, n-gram LMs and RNNLMs are usually combined using a context independent, fixed weighting based linear interpolation in state-of-the-art ASR systems [1, 2, 3, 6, 7]. The same approach was previously used to combine multiple n-gram LMs trained, for example, on a diverse collection of data sources. In order to reduce the mismatch between the interpolated LM and the task of interest, interpolation weights may be tuned by minimizing the perplexity on some held-out data [8, 9, 10, 11, 12]. These interpolation weights indicate the "usefulness" of individual component LMs for a particular task.

In order to fully exploit the locally varying complementary attributes among component LMs during interpolation, a more general history context dependent form of interpolation can be used to combine n-gram LMs [13]. A similar local variation of probabilistic contribution from n-gram LMs and RNNLMs across different contexts during interpolation was also reported in previous research [14]. The perplexity analysis over n-gram LMs and RNNLMs in [14] suggests such variation is heavily correlated with the underlying context resolution of component n-gram LMs. For example, RNNLMs assign higher probabilities when the n-gram LMs' context resolution is significantly reduced via the back-off recursion to a lower order, and reversely when a longer history context can be modelled by the n-gram LMs without using back-off. Inspired by these findings, a back-off based compact representation of n-gram dependent interpolation weights is investigated in this paper. This approach allows robust weight parameter estimation on limited data. Experiments are conducted on the three tasks with varying amounts of training data. Small and consistent improvements in both perplexity and WER were obtained using the proposed interpolation approach over the baseline fixed weighting based linear interpolation.

This paper is organized as follows. Section 2 gives a brief review of n-gram LMs and RNNLMs. Section 3 presents the conventional linear interpolation between n-gram LMs and RNNLMs. Section 4 proposes a novel back-off based interpolation between n-gram LMs and RNNLMs. Experimental results are reported on three tasks with different amount of training data in Section 5. Conclusions are drawn and possible future work is discussed in Section 6.

2. LANGUAGE MODELS

Statistical language models (LMs) assign a probability to a given sentence $\mathcal{W} = <w_1, w_2, ..., w_N>$.

$$P(\mathcal{W}) = \prod_{i=1}^{N} P(w_i|h_1^{i-1})$$

Xie Chen is supported by Toshiba Research Europe Ltd, Cambridge Research Lab. The research was also supported by EPSRC grant EP/I031022/1 (Natural Speech Technology) program and in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The paper does not necessarily reflect the position or the policy of US Government and no official endorsement should be inferred. Supporting data for this paper is available at the https://www.repository.cam.ac.uk/handle/1810/251277 data repository.

where $h_1^{i-1} =< w_1, ..., w_{i-1} >$ denotes the history context for word w_i. The probability distribution given any history context h_1^{i-1} is required to satisfy a positive and sum-to-one constraint

$$\sum_w P(w|h_1^{i-1}) = 1. \quad (1)$$

In addition to extrinsic LM performance evaluation metrics that are based on, for example, word error rates for speech recognition tasks, the generalization performance of language models can also be evaluated using the perplexity measure. This is defined as

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\ln P(w_i|h_1^{i-1})\right). \quad (2)$$

In general, a language model with lower perplexity generalise better when used to predict unseen data.

2.1. n-gram LMs

Back-off n-gram LMs have been the dominant form of statistical language models (LMs) during the last few decades due to their simple model structures, efficient parameter estimation techniques and discounting algorithms to improve robustness. Good generalization performance can be obtained using back-off n-gram LMs when large quantities of training data are available. Under a Markov chain assumption, the probability of the current word being predicted only depends on preceding $N-1$ words. This is given by

$$P(w_i|h_1^{i-1}) = P_{\text{NG}}(w_i|h_{i-N+1}^{i-1}). \quad (3)$$

The n-gram LM probabilities $P_{\text{NG}}(w_i|h_{i-N+1}^{i-1})$ are computed using the following back-off recursion

$$P_{\text{NG}}(w_i|h_{i-N+1}^{i-1}) = \begin{cases} P_{\text{NG}}(w_i|h_{i-N+1}^{i-1}) & \text{if } C(w_i, h_{i-N+1}^{i-1}) > C^N \\ \alpha(h_{i-N+1}^{i-1}) P_{\text{NG}}(w_i|h_{i-N+2}^{i-1}) & \text{otherwise} \end{cases} \quad (4)$$

Where C^N is the N^{th} order count cut-off, and $C(w_i, h_{i-N+1}^{i-1})$ is the frequency count of a particular n-gram $<w_i, h_{i-N+1}^{i-1}>$ in the training corpus. The history dependent normalisation term $\alpha(h_{i-N+1}^{i-1})$ ensures $P_{NG}(w_i|h_{i-N+1}^{i-1})$ be a valid probability distribution.

A key part of the statistical language modelling problem for many applications including speech recognition is to model the long-distance context dependencies in natural languages. Directly modelling long-span history contexts using n-gram LMs can in general lead to a severe data sparsity problem. In order to improve robustness, in state-of-the-art speech recognition systems, n-grams LMs are often constructed using sophisticated parameter smoothing techniques represented by, for example, modified KN smoothing [15].

2.2. Recurrent Neural Network LMs

In contrast to n-gram LMs, recurrent neural network LMs [1] represent ine fixed weighting based linear interpolation. the full, non-truncated history $h_1^{i-1} =< w_1, \ldots, w_{i-1} >$ for word w_i using a 1-of-k encoding of the previous word w_{i-1} and a continuous vector $\boldsymbol{v}_{i-2}$ for the remaining context. For an empty history, this is initialised, for example, to a vector of all ones. An out-of-vocabulary (OOV) input node can also be used to represent any input word not in the chosen recognition vocabulary. The topology of the recurrent neural network used to compute LM probabilities $P_{\text{RNN}}(w_i|w_{i-1}, \boldsymbol{v}_{i-2})$ consists of three layers. The full history vector, obtained by concatenating w_{i-1} and $\boldsymbol{v}_{i-2}$, is fed into the input layer. The hidden layer compresses the information from these two inputs and computes a new representation $\boldsymbol{v}_{i-1}$ using a sigmoid activation to achieve non-linearity. This is then passed to the output layer to produce normalized RNNLM probabilities using a softmax activation, as well as recursively fed back into the input layer as the "future" remaining history to compute the LM probability for the following word $P_{\text{RNN}}(w_{i+1}|w_i, \boldsymbol{v}_{i-1})$. An example RNNLM architecture with an unclustered, full output layer is shown in Figure 1.

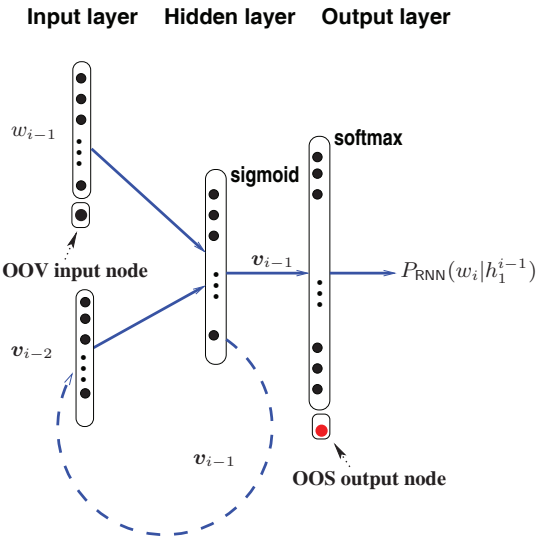

Fig. 1. *RNNLM with an full output layer and OOS nodes.*

RNNLMs can be trained using an extended form of the standard back propagation algorithm, back propagation through time (BPTT) [16], where the error is propagated through recurrent connections back for a specific number of time steps, for example, 4 or 5 [2]. This allows RNNLMs to keep information for several time steps in the hidden layer. To reduce the computational cost, a shortlist [17, 18] based output layer vocabulary limited to the most frequent words can be used. To reduce the bias to in-shortlist words during RNNLM training and improve robustness, an additional node is added at the output layer to model the probability mass of out-of-shortlist (OOS) words [19, 20, 21].

RNNLM training is computationally expensive. This practical issue limits the quantity of data and the number of possible application areas for RNNLMs. In order to solve this problem, recently there has increasing research interest in deriving efficient parallel training algorithms for RNNLMs [22, 23, 24, 25]. In particular, RNNLMs with a full output layer were efficiently trained on a graphics processor unit (GPU) using a spliced sentence bunch based parallel training algorithm on large amounts of data in [25]. A modified version of the RNNLM toolkit [26] supporting the above GPU based parallel RNNLM training method and the full output RNNLM architecture shown in Figure 1 is used in this paper. RNNLM training is more time-consuming compared to n-gram LMs. In order to improve efficiency, for state-of-art ASR tasks, n-gram LMs are often trained on a significantly larger quantities of data, while RNNLMs are trained on smaller amounts of in-domain data. In this work, n-gram LMs and RNNLMs are trained on the same data to allow a fair comparison.

As RNNLMs use a vector representation of full histories, they are mostly used for N-best list rescoring. For more efficient lattice rescoring using RNNLMs, appropriate approximation schemes, for example, based on clustering among complete histories proposed in [21] can be used.

3. LANGUAGE MODEL INTERPOLATION

Methods to combine multiple language models had been studied and compared in [27, 13, 28]. Most of these techniques are investigated on n-gram LMs and their derivations, such as topic based n-gram LM and cached based n-gram LM. RNNLMs are inherently different from n-gram LMs in terms of their generalisation patterns. For this reason, RNNLMs are usually linearly interpolated with n-gram LMs to obtain both a good context coverage and strong generalisation [1, 3, 17, 18, 19, 20]. The interpolated LM probability is given by

$$P(w_i|h_1^{i-1}) \;=\; \lambda P_{\mathsf{NG}}(w_i|h_1^{i-1}) + (1-\lambda)P_{\mathsf{RN}}(w_i|h_1^{i-1}) \quad (5)$$

λ is the global weight of the n-gram LM distribution $P_{\mathsf{NG}}(\cdot)$, which can be optimized using the EM algorithm on a held-out set.

4. BACK-OFF BASED LM INTERPOLATION

4.1. Generalized LM Interpolation Using Weight Clustering

As discussed in sections 1 and 3, in order to fully exploit the complementary attributes between n-gram LMs and RNNLMs that vary among individual n-gram contexts, a more general form of linear probability interpolation between the two based on n-gram dependent weights can be considered. However, this approach requires a large number of interpolation weight parameters to be robustly estimated and therefore leads to a severe data sparsity problem when given limited data. A general solution to handle this problem to share weights within groups of contexts where the contribution from n-gram LMs and RNNLMs (represented by the interpolation weights) are similar. Using this approach a more compact representation of the n-gram dependent interpolation weights can be derived. This allows weight parameters to be robustly estimated on limited data. The fixed weighting based linear interpolation in equation (5) is thus modified as

$$\begin{aligned} P(w_i|h_1^{i-1}) \;=\; & \frac{1}{Z(h_1^{i-1})}\Big(\lambda^{(\mathsf{NG})}_{\Phi\left(w_i,h_1^{i-1}\right)} P_{\mathsf{NG}}(w_i|h_1^{i-1}) \quad (6)\\ & +\lambda^{(\mathsf{RN})}_{\Phi\left(w_i,h_1^{i-1}\right)} P_{\mathsf{RN}}(w_i|h_1^{i-1})\Big) \end{aligned}$$

where the n-gram dependent interpolation weights $\lambda^{(NG)}_{\Phi\left(w_i,h_1^{i-1}\right)}$ and $\lambda^{(RN)}_{\Phi\left(w_i,h_1^{i-1}\right)}$ are positive values and shared using an n-gram clustering function $\Phi(\cdot)$. A normalisation term $Z(h_1^{i-1})$ is also required to ensure the interpolated LM probabilities to be valid. This term is computed as

$$\begin{aligned} Z(h_1^{i-1}) \;=\; & \sum_{w'}\Big(\lambda^{(\mathsf{NG})}_{\Phi\left(w',h_1^{i-1}\right)} P_{\mathsf{NG}}(w'|h_1^{i-1}) \quad (7)\\ & +\lambda^{(\mathsf{RN})}_{\Phi\left(w',h_1^{i-1}\right)} P_{\mathsf{RN}}(w'|h_1^{i-1})\Big) \end{aligned}$$

The above form of interpolation based on n-gram weight classing is illustrated in Figure 2. Usually, the interpolation weights of n-gram LM and RNNLM satisfy $\lambda^{(\mathsf{NG})}_{\Phi\left(w',h_1^{i-1}\right)}+\lambda^{(\mathsf{RN})}_{\Phi\left(w',h_1^{i-1}\right)}=1$. By definition, the standard fixed weight based linear interpolation in equation (5) is subsumed by the more general form of linear interpolation in equation (6), and is equivalent to assigning all n-gram contexts to a single class and fixed interpolation weights are used.

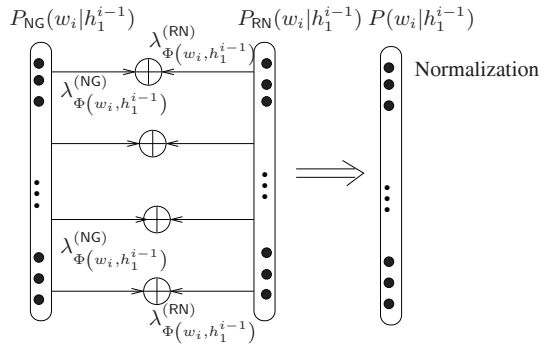

Fig. 2. *n-gram dependent interpolation of n-gram LM and RNNLM.*

4.2. Interpolation using back-off for weight clustering (n-gram LM $\oplus$ RNNLM)

A central part of the n-gram class dependent interpolation approach given in equation (6) is to derive an appropriate form of n-gram context classing $\Phi(\cdot)$. For the particular form of interpolation between back-off n-gram LMs and RNNLMs considered in this paper, a suitable weight classing scheme is expected to reflect the variation of the probabilistic contribution from these two component LMs. In previous research it was found that such variation is heavily correlated with the n-gram LM's underlying context resolution. This is represented by the highest available n-gram order obtained through the back-off recursion in n-gram LM in equation (4).

Such correlation is analyzed again in this paper on the Penn TreeBank (PTB) corpus. A detailed breakdown of the perplexity performance of the baseline 5-gram LM, RNNLM and their linear interpolation over different n-gram context groups of varying back-off orders on the PTB test data is shown in table 1. The 5-gram LM outperformed the RNNLM in terms of the overall perplexity. As expected, significant perplexity reduction was further obtained using standard optimized linear interpolation (3rd line in table 1). A large variation in the 5-gram LM's contribution characterized by the rank ordering in perplexity against the RNNLM over different back-off orders is also clearly shown in table 1). This is due to the fact that n-gram LMs and RNNLMs employ inherently different mechanism to acquire generalization. n-gram LMs are more powerful in predicting the probabilities of frequently occurring n-grams using higher order context modelling, while RNNLMs' strength lie in their ability to predict rare events.

The above example analysis suggests back-off order can provide a highly compact representation of n-gram level varying probabilistic contribution from the back-off n-gram LMs and RNNLMs. As the interpolation weights are clustered into a small number of classes associated with the back-off orders, the weight parameters can be robustly estimated on a small amount of held-out data. The associated

Table 1. Perplexity performance of baseline 5-gram LM, RNNLM and their linear interpolation over varying back-off n-gram orders on PTB test set.

LM	n-gralm LM back-off level					Overall
	1g	2g	3g	4g	5g	
#words	15594	33646	19655	9502	4033	82430
5G	9157.3	198.0	26.4	8.3	2.5	141.5
RNN	4633.7	183.4	38.7	17.9	6.0	150.8
5G+RNNLM	4697.6	161.6	26.9	9.4	3.0	118.3

interpolation weight classing is thus computed as

$$\Phi\Big(w_i, h_1^{i-1}\Big) = \Phi_{\mathsf{NG}}\left(w_i, h_{i-N+1}^{i-1}\right) = \begin{cases} N & \text{if } <w_i, h_{i-N+1}^{i-1}> \in \mathcal{G}_{\mathsf{NG}} \\ \Phi_{\mathsf{NG}}\left(w_i, h_{i-N+2}^{i-1}\right) & \text{otherwise} \end{cases} \quad (8)$$

where $\mathcal{G}_{\mathsf{NG}} = \{..., <w', h'>, ...\}$ contains all the unique observed n-gram contexts that the n-gram LM $P_{\mathsf{NG}}(\cdot)$ models. However, the optimization of interpolation weights is not easy due to the normalisation term $Z(h_1^{i-1})$ in Equation 7. Stochastic gradient descent is applied for optimization on a held-out set and the condition $\lambda^{(\mathsf{NG})}_{\Phi_{\mathsf{NG}}(w_i,h_1^{i-1})} + \lambda^{(\mathsf{RN})}_{\Phi_{\mathsf{NG}}(w_i,h_1^{i-1})} = 1$ is retained for each back-off level during optimization. This form of interpolation will be denoted as n-gram LM $\oplus$ RNNLM.

4.3. Back-off based interpolation with rescaling (n-gram LM $\otimes$ RNNLM)

The n-gram class dependent interpolation approach given in equation (6) requires a normalisation term $Z(h_1^{i-1})$ to be computed for each distinct history context over multiple weight classes. As such term is also dependent on the interpolation weights, a direct optimization of the weight parameters by maximising the interpolated LM probabilities in equation (6) is a non-trivial problem. Computationally expensive numerical optimization methods are required.

In order to improve efficiency, an alternative novel form of n-gram class dependent interpolation between back-off LMs and RNNLMs is considered in this paper. This is given by

$$\begin{aligned} P_{\mathsf{BOInt}}(w_i|h_1^{i-1}) &= \lambda_{\Phi\left(w_i,h_1^{i-1}\right)} P_{\mathsf{NG}}(w_i|h_1^{i-1}) \\ &+ \Big(1 - \lambda_{\Phi\left(w_i,h_1^{i-1}\right)}\Big)\, \beta_{\Phi\left(w_i,h_1^{i-1}\right),h_1^{i-1}} P_{\mathsf{RN}}(w_i|h_1^{i-1}) \end{aligned} \quad (9)$$

where the n-gram context class and history dependent normalisation term $\beta_{\Phi\left(w_i,h_1^{i-1}\right),h_1^{i-1}}$ is independent of interpolation weight parameters and computed as below.

$$\beta_{\Phi\left(w_i,h_1^{i-1}\right),h_1^{i-1}} = \frac{\sum_{w',\Phi(w',h_1^{i-1})=\Phi\left(w_i,h_1^{i-1}\right)} P_{\mathsf{NG}}(w'|h_1^{i-1})}{\sum_{w',\Phi(w',h_1^{i-1})=\Phi\left(w_i,h_1^{i-1}\right)} P_{\mathsf{RN}}(w'|h_1^{i-1})} \quad (10)$$

Recalling the general form of interpolation in Equation 6, Equation 9 is a specific case of it under the following conditions,

$$\begin{aligned} \lambda^{(\mathsf{NG})}_{\Phi\left(w_i,h_1^{i-1}\right)} &= \lambda_{\Phi\left(w_i,h_1^{i-1}\right)} \\ \lambda^{(\mathsf{RN})}_{\Phi\left(w_i,h_1^{i-1}\right)} &= \Big(1 - \lambda_{\Phi\left(w_i,h_1^{i-1}\right)}\Big)\, \beta_{\Phi\left(w_i,h_1^{i-1}\right),h_1^{i-1}} \\ Z(h_1^{i-1}) &= 1 \end{aligned}$$

As the above normalisation term is no longer dependent on the interpolation weights, weight parameters associated with different classes can be optimized independently of each other using the conventional EM algorithm on held-out data. During evaluation, this normalisation term can be computed for each pairing of the underlying weight class and history context only once and cached for efficiency. In common with the interpolation approach given in equation (6), the form of interpolation in equation (9) also requires a suitable form of interpolation weight class assignment among different n-gram contexts. The back-off based interpolation weight classing given in equation (8) is used.

The resulting back-off based interpolation given in equation (9) retains the probability mass of all n-grams sharing a common history and the same back-off based weight class. This probability mass is then be redistributed using the RNNLM distribution during interpolation. In this process, potential bias to the n-gram LM distribution may be introduced in the final interpolated LM probabilities. In order to address this issue, it is possible to further improve generalisation performance by combining the interpolated LM probabilities obtained using equation (9) with RNNLMs using the conventional fixed weighting based interpolation.

$$P(w_i|h_1^{i-1}) = \lambda P_{\mathsf{BOInt}}(w_i|h_1^{i-1}) + (1-\lambda) P_{\mathsf{RN}}(w_i|h_1^{i-1}) \quad (11)$$

The back-off class dependent interpolation weights $\left\{\lambda_{\Phi\left(w_i,h_1^{i-1}\right)}\right\}$ and the top level linear interpolation weight λ can be optimized iteratively using the EM algorithm on a held-out set. This form of interpolation will be denoted as n-gram LM $\otimes$ RNNLM + RNNLM.

5. EXPERIMENTS

Experiments are conducted on three tasks with different amounts of training data to show the effect of back-off based interpolation. First, the Penn TreeBank (PTB) Corpus is used for experiment to validate the previous findings reported in [14]. The 860k word PTB training data and a 10k vocabulary were used. A development data set of 70k words was used parameter tuning. A separate 79k word test test was used for performance evaluation. The perplexity (PPL) results of the 5-gram LM and RNNLM are shown in Table 2. The PPL scores the two LMs over different context groups associated with varying back-off n-gram orders are shown in the first two rows. These results were previously presented and discussed in table 1 in Section 4.2. The third line (5G+RNN) shows the PPL score breakdown of the final linear interpolated LM. The linear interpolation weight λ was perplexity optimized on the development set. According to these results, the conventional form of linear interpolation gave good generalisation performance on each back-off order context groups via a simple probability averaging. The overall PPL was reduced from 141.5 to 118.3. In particular, this standard fixed weighting based interpolated LM gave significant improvements in PPL for the group of contexts where the 5-gram LM backed off to 1-gram.

The fourth line (5G$\oplus$RNN) gives the results of the first back-off based interpolation method introduced in Section 4.2, the interpolation weighs were optimized with stochastic gradient descent. It gave a slight overall PPL improvement. The fifth line (5G$\otimes$RNN) presents the results of the back-off based interpolation approach of Section 4.3. As discussed, this form of back-off based interpolation retains the n-gram LM's probability mass of all n-grams sharing a common history and the same back-off order based weight class, and re-distributes it using the RNNLM distribution during interpolation. It could be seen from Table 2 that the PPL score was improved on each back-off level compared to the baseline 5-gram LM.

Table 2. PPL results on test set of PTB corpus

LM	n-gralm LM back-off level					Overall
	1	2	3	4	5	
#words	15594	33646	19655	9502	4033	82430
5G	9157.3	198.0	26.4	8.3	2.5	141.5
RNN	4633.7	183.4	38.7	17.9	6.0	150.8
5G+RNNLM	4697.6	161.6	26.9	9.4	3.0	118.3
5G$\oplus$RNN	4568.1	163.0	27.4	9.1	2.9	117.8
5G$\otimes$RNN	5472.1	170.0	24.3	7.9	2.4	117.6
RNN$\otimes$5G+RNN	5230.7	167.8	24.7	8.1	2.5	117.0

The interpolation weight λ_n on each back-off level was efficiently optimized independently via the EM algorithm on the development data. The optimal interpolation weights $\{\lambda_n\}$ for the 5-gram LM were $\{0.25, 0.4, 0.5, 0.55, 0.55\}$ for varying back-off levels from 1 to 5. As expected, a general trend could be found that the n-gram weight increases with the back-off order. The back-off based interpolated LM probabilities could be further linearly interpolated with the RNNLM (with a weighting 0.9:0.1) using equation (11). This gave further small improvements in perplexity.

The next experiment was conducted on the BABEL corpus (i.e. IARPA-babel202b-v1.0d) and used Full Language Pack (FLP) of the Swahili language. A 3-gram LM (3glm) with slight pruning and RNNLM were both trained on 290K words of text data [1]. The test set includes 52K words. The vocabulary size is 24K. All vocabulary words were used in RNNLM input and output word listst during training. A total of 200 hidden units were used. RNNLMs were trained on GPU as described in [25]. The PPL and WER results are shown in Table 3. A pattern similar to that observed on the PTB task in Table 2 was found. Standard linear interpolation reduced the overall PPL score by 7% relative compared with the 3-gram LM. A detailed analysis on each back-off level showed that linear interpolation improved the PPL score by 25% relative on the words where the 3-gram LM backs off to 1-gram, while no improvements were obtained for the other two back off levels. The back-off based interpolation by simply clustering (3G$\oplus$RNN) provided slight PPL reduction and obtained the same WER as linear interpolation. The back-off based interpolation (3G$\otimes$RNN) reduced the PPL consistently on each back-off level compared with the 3-gram LM. A small overall PPL reduction was also obtained over the conventional fixed weight based linear interpolation. The optimal interpolation weights assigned to the 3-gram LM were (0.25, 0.6, 0.65) for the back-off levels from 1-gram to 3-gram.

Table 3. PPL and WER results on Swahili for Babel

LM	PPL				WER
	n-gram LM back-off level			Overall	
	1	2	3		
#words	19687	28355	7156	55198	
3G	3510.4	107.1	19.0	297.2	47.3
RNN	2387.3	145.6	29.9	321.6	-
3G+RNNLM	2618.3	107.9	21.5	273.0	46.8
3G$\oplus$RNN	2602.6	107.6	21.6	272.2	46.8
3G$\otimes$RNN	2933.9	102.0	18.7	271.3	46.9
3G$\otimes$RNN+RNN	2850.4	103.0	19.2	270.7	46.7

ASR experiments were then conducted on the same BABEL task. The acoustic models were trained on 46 hours of speech. Tandem and hybrid DNN systems were trained separately. A frame level joint decoding was then applied to combine the acoustic scores of the two systems [29]. The baseline 3-gram LM was used in the first decoding stage for lattice generation. N-best (N=50) rescoring was then applied using the interpolation between the RNNLM and 3-gram LM. The word error rate (WER) results are shown in Table 3. The baseline 4-gram gave a WER score of 47.3%. Standard linear interpolation gave an absolute 0.5% WER reduction. the back-off based interpolation gave a comparable WER score of 46.9%. A further linear interpolation using equation (11) between the back-off based interpolated LM and the RNNLM (with a weighting 0.9:0.1) gave the lowest WER of 46.7% in the table.

The previous experiments were conducted on a relatively small amount of training data. In the next experiment a much larger training set based on the BBC Multi-Genre Broadcast (MGB) challenge task was used[2]. 650M words of text data were used in the baseline 4-gram LM (4glm) and RNNLM training. The hybrid DNN acoustic model was trained on 700 hours of data. A 64K vocabulary was used. A total of 500 hidden nodes were used in the RNNLM. A 46K input shortlist and 40K output shortlist were used in RNNLM training. The results are shown in Table 4. The complementary attributes of 4-gram LM (4glm) and the RNNLM on each back-off level were consistent with the previous two tasks. There was only 3.6% of all n-gram requests that back off to 1-gram due to the large amount of training data being used and low pruning threshold. In common with the previous two tasks, RNNLM was found to perform better than 4-gram LM when the latter backs off to 1-gram or 2-gram probabilities, while vice versa when it retained a 3-gram or 4-gram modelling resolution. The baseline linear interpolation gave a significant overall reduction in perplexity. On each back-off level, the reduction in perplexity increases when the back-off level decreases. Again the back-off based interpolation with simply clustering (4G$\oplus$RNN) gave slight PPL improvement and the same WER compared to linear interpolation. The back-off based interpolation with rescaling (4G$\otimes$RNN) slightly outperformed the conventional linear interpolation. In terms of WER results, The linear interpolation reduce the WER by 0.7% absolutely. 4G$\otimes$RNN and 4G$\otimes$RNN+RNN gave a small further reduction of of 0.1% absolute and gave an overall improvement 0.8% absolute over the baseline 4-gram LM.

Table 4. PPL and WER results on MGB task

LM	PPL					WER
	n-gram LM back-off level				Overall	
	1	2	3	4		
#words	7362	60578	78528	56291	202759	
4G	18731.7	733.2	76.0	11.7	108.7	26.2
RNN	5868.1	564.3	79.0	21.8	116.2	-
4G+RNN	6782.0	560.3	68.2	13.4	96.3	25.5
4G$\oplus$RNN	6440.7	563.0	68.7	12.7	95.1	25.5
4G$\otimes$RNN	7145.7	593.5	70.0	11.5	94.9	25.4
4G$\otimes$RNN+RNN	6800.4	584.2	69.5	11.8	94.7	25.4

6. CONCLUSIONS AND FUTURE WORK

In order to exploit the complementary features among n-gram LMs and RNNLMs, a standard form of linear interpolation based on fixed weighs is widely used to combine them. Motivated by their inherently different generalization patterns that are correlated with the variation of the underlying n-gram LM context resolution, a novel back-off based compact representation of n-gram dependent interpolation weights is proposed in this paper. The proposed technique

[1] A 4-gram LM gave no further improvements of ASR performance given the small amount of training data

[2] The detail of MGB challenge could be found from http://www.mgb-challenge.org/

allows the interpolation weights shared at each back-off level to be estimated both efficiently and robustly. Experimental results on three tasks of varying amounts of training data show that the proposed back-off based linear interpolation between n-gram LMs and RNNLMs provided a simple but powerful way to combine them. Small but consistent improvements in terms of both perplexity and WER reductions were obtained over the conventional fixed weighting based linear interpolation. Alternative interpolation methods to further improve the combination between RNNLMs and n-gram LMs will be investigated in future research.

7. REFERENCES

[1] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model.," in *Proc. ISCA Interspeech*, 2010.

[2] Tomas Mikolov, Stefan Kombrink, Lukas Burget, J.H. Cernocky, and Sanjeev Khudanpur, "Extensions of recurrent neural network language model," in *Proc. ICASSP*. IEEE, 2011.

[3] Martin Sundermeyer, Ilya Oparin, Jean-Luc Gauvain, Ben Freiberg, Ralf Schluter, and Hermann Ney, "Comparison of feedforward and recurrent neural network language models," in *Proc. ICASSP*, 2013.

[4] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu, "Recurrent neural networks for language understanding.," in *Proc. ISCA Interspeech*, 2013.

[5] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio, "Investigation of recurrent neural network architectures and learning methods for spoken language understanding.," in *Proc. ISCA Interspeech*, 2013.

[6] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson, "One billion word benchmark for measuring progress in statistical language modeling," Tech. Rep., Google, 2013.

[7] Xie Chen, Mark Gales, Kate Knill, Catherine Breslin, Langzhou Chen, K.K. Chin, and Vincent Wan, "An initial investigation of long-term adaptation for meeting transcription," in *Proc. ISCA Interspeech*, 2014.

[8] Frederick Jelinek and Robert Mercer, "Interpolated estimation of markov source parameters from sparse data.," in *Proc. Workshop on Pattern Recognition in Practice*, 1980.

[9] Reinhard Kneser and Hermann Ney, "Improved clustering techniques for class-based statistical language modelling," in *Proc. EUROSPEECH*, 1993.

[10] Rukmini Iyer and Mari Ostendorf, "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 30–39, 1999.

[11] Ronald Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech & Language*, vol. 10, no. 3, pp. 187–228, 1996.

[12] Philip Clarkson and Anthony Robinson, "Language model adaptation using mixtures and an exponentially decaying cache," in *Proc. ICASSP*. IEEE, 1997.

[13] Xunying Liu, Mark Gales, and Phil Woodland, "Use of contexts in language model interpolation and adaptation," *Computer Speech & Language*, pp. 301–321, 2013.

[14] Ilya Oparin, Martin Sundermeyer, Hermann Ney, and Jean-Luc Gauvain, "Performance analysis of neural networks in combination with n-gram language models," in *Proc. ICASSP*. IEEE, 2012.

[15] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*. IEEE, 1995.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, *Learning representations by back-propagating errors*, MIT Press, Cambridge, MA, USA, 1988.

[17] Holger Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[18] Ahmad Emami and Lidia Mangu, "Empirical study of neural network language models for Arabic speech recognition," in *ASRU, IEEE Workshop on*. IEEE, 2007.

[19] Junho Park, Xunying Liu, Mark Gales, and Phil Woodland, "Improved neural network based language modelling and adaptation," in *Proc. ISCA Interspeech*, 2010.

[20] Hai-Son Le, Ilya Oparin, Alexandre Allauzen, J Gauvain, and François Yvon, "Structured output layer neural network language models for speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 197–206, 2013.

[21] Xunying Liu, Yongqiang Wang, Xie Chen, Mark Gales, and Phil Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *Proc. ICASSP*. IEEE, 2014.

[22] Yangyang Shi, Mei-Yuh Hwang, Kaisheng Yao, and Martha Larson, "Speed up of recurrent neural network language models with sentence independent subsampling stochastic gradient descent," in *Proc. ISCA Interspeech*, 2013.

[23] Zhiheng Huang, Geoffrey Zweig, Michael Levit, Benoit Dumoulin, Barlas Oguz, and Shawn Chang, "Accelerating recurrent neural network training via two stage classes and parallelization," in *ASRU, IEEE Workshop on*. IEEE, 2013.

[24] Boxun Li, Erjin Zhou, Bo Huang, Jiayi Duan, Yu Wang, Ningyi Xu, Jiaxing Zhang, and Huazhong Yang, "Large scale recurrent neural network on gpu," in *Neural Networks International Joint Conference*. IEEE, 2014.

[25] Xie Chen, Yongqiang Wang, Xunying Liu, Mark Gales, and Phil Woodland, "Efficient GPU-based training of recurrent neural network language models using spliced sentence bunch," in *Proc. ISCA Interspeech*, 2014.

[26] Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukas Burget, and Jan Cernocky, "Recurrent neural network language modeling toolkit," in *ASRU, IEEE Workshop*, 2011.

[27] Simo Broman and Mikko Kurimo, "Methods for combining language models in speech recognition.," in *Proc. ISCA Interspeech*, 2005.

[28] Bo-June Hsu, "Generalized linear interpolation of language models," in *ASRU. IEEE Workshop*, 2007, pp. 136–140.

[29] Haipeng Wang, Anton Ragni, Mark Gales, Kate Knill, Phil Woodland, and Chao Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Proc. ISCA Interspeech*, 2015.