STACKED NEURAL NETWORKS WITH PARAMETER SHARING FOR MULTILINGUAL LANGUAGE MODELING

Banriskhem K. Khonglah, Srikanth Madikeri, Navid Rekabsaz, Nikolaos Pappas, Petr Motlicek, Hervé Bourlard

Idiap Research Institute, Martigny, Switzerland
{banriskhem.khonglah, srikanth.madikeri, navid.rekabsaz, nikolaos.pappas, petr.motlicek, herve.bourlard}@idiap.ch

ABSTRACT

Neural language models are useful in Automatic Speech Recognition (ASR) due to their superior re-scoring capabilities over N-gram language models. Recently, multilingual neural language modeling based on recurrent neural networks has gained attraction for the same purpose. Multilingual models aim to transfer knowledge across languages especially to the ones which have limited domain training data. However, even though recurrent neural networks improve perplexity scores in language modeling when trained in a multilingual manner, they still have not been shown helpful for re-ranking in ASR. Here, we propose multiple stacked neural networks with a layer on top across languages trained in a multilingual setting which improve over previous monolingual and multilingual baselines in both language modeling and ASR. Our best model has a Time Delay Neural Network (TDNN) specific to a particular language at the bottom layer, while its top layer is a Long Short Term Memory (LSTM) shared across multiple languages. The former aims to capture the characteristics of the individual languages, while the latter aims to capture the common sentence structure across languages. We evaluate our models on four BABEL languages in terms of perplexity on language modeling and in terms of word error rate on ASR.

Index Terms— Multilingual language modeling, Stacked neural networks, Re-ranking for speech recognition

1. INTRODUCTION

Neural Language Models (NLM) which are generally used for re-ranking hypotheses in Automatic Speech Recognition (ASR) require domain-specific training data, such as conversational speech data, for a particular language in order to work properly. However, the availability of such domain-specific data can be limited, particularly for low-resource languages. Multilingual neural network language models which share across multiple languages, aim to address such data sparsity issues [1]. In general, such multilingual models have language-specific as well as language-independent parameters which are shared among all languages. Parameter sharing across multiple languages may act as implicit regularization for the neural language models involved, especially for languages with insufficient amount of data, due to the knowledge transfer that is happening across languages.

Multilingual training of neural networks has grown in the last few years on acoustic modeling for ASR [2–4], as well as for language processing tasks such as document classification [5] and machine translation [6]. Ragni et al. [1] proposed a multilingual neural language model which is related to previous work on bottleneck features for multilingual acoustic modeling in ASR [3]. It comprises of a Recurrent Neural Network (RNN) with one layer where the weights of the hidden layer of the RNN are shared across the languages, while the input and output layers are language-specific. Their work is different from the acoustic model multilingual training [3],because the inputs for the acoustic model are fixed-dimensional feature vectors whereas the inputs to the language model are index representations of different words learned end-to-end. Furthermore, the multilingual model in [1] is further fine-tuned on each language by separately retraining the model per language. In fact, their multilingual approach effectively initializes the weights of the RNN layer, helping the models per language to effectively converge and hence achieve better results. Lastly, the input and output layers in [1] contain the vocabulary lists of all languages, meaning that the loss is computed over the entire vocabularies of languages. This approach however can cause bias when computing Softmax in the output layer for a given language, since unrelated vocabularies (the ones in other languages) are considered in the normalization factor of Softmax for the vocabularies of that

The research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

language.

In this work, we pursue the direction of [1], by proposing a state-of-the-art approach for multilingual neural language modeling for re-ranking hypotheses in ASR. Our proposed multilingual architecture consists of a stacked neural network model with two layers, where the first layer is language-specific and the top is shared across multiple languages. In addition, in contrast to [1], every language has separate input and output layers and hence a separate loss function. The overall loss in our proposed approach is the weighted sum of per-language loss values, used to optimize the whole network through back-propagation (details in Section 3).

Various forms of recurrent neural network models have been widely used for language modeling [7–10], among which, Long Short Term Memory (LSTM) networks demonstrate the most strong and reliable performance [11]. As shown recently, recurrent neural networks are also capable of capturing the common sentence structures among various languages [12]. Recently, the feed-forward neural networks, especially in Time Delay Neural Network (TDNN) models, have also shown competitive performance compared to LSTMs, since the infinite memory capacity of recurrent networks is actually absent in practice [13–15]. In the light of these studies, we explore the combination of these two architectures (TDNN and LSTM) for multilingual language modeling, one for the language-specific and the other for the language-independent layer.

Our experimental results show noticeable improvement of our proposed multilingual models in comparison to monolingual models as well as previously proposed multilingual approaches for two out of the four languages used for training, both in terms of perplexity values for language modeling and Word Error Rate (WER) for ASR systems. Even though the improvements brought by our multilingual model are not observed in every language used in our experiments, the WER is always at least as good or better than the performance based on monolingual models. Furthermore, we also observe a link between improvement in perplexity and WER for Tagalog and Swahili which is a promising finding. Our best performing architecture uses a TDNN for the language-specific layer to capture the particular characteristics of individual languages, and a LSTM for the shared layer to capture the common sentence structures across languages. We show that our proposed method of combining the networks (assuming the training principle remains the same) enables a complementary capture of information by the two networks for multilingual language modeling. As opposed to [1], our proposed model does not require fine-tuning for obtaining gains in performance, saving the re-training phase on each language, but could perhaps benefit from it separately.

The paper is organized as follows. Section 2 provides background on language models in ASR as well as the multilingual acoustic model. The proposed multilingual language model is described in Section 3. The results are presented and discussed in Section 4, and Section 5 concludes the study.

2. BACKGROUND

2.1. Language Modeling in ASR

Language modeling in ASR is typically based on the N-grams which involve estimating the probability based on counting. Later smoothing techniques were introduced to make the N-grams more robust to zero counts [16]. Given the recent advances in language modeling using neural networks, one way of exploiting them in ASR is by re-scoring the N-best list of hypotheses obtained from the decoding based on N-grams [7]. This is usually called the second pass decoding. There are also attempts to use the neural networks for first pass decoding. In [17], the RNN is used as a generative model to generate text. The N-gram LMs are then trained based on the generated text and finally used for decoding. In [18], RNN LM histories are discretized to create Weighted Finite State Transducers (WFST). A probability-based conversion has been explored in [19], and this method involves the extraction of N-grams but the count based probabilities are replaced by the RNN-LM probabilities. In the present study, we use the second pass decoding, where the neural language model is used to re-score the N-best hypothesis generated by the N-gram language model.

2.2. Multilingual Acoustic Modeling

To train acoustic models for ASR with limited amount of data, it is helpful to augment data to avoid sub-optimal convergence. There are multiple ways to augment data. One method is to create multiple copies of the training data by adding noise or varying speed and volume. Another efficient method is to train a multi-task network. In a multi-task network, it is possible to combine several low-resource languages to eliminate the data insufficiency problem. In [20], 19 languages from the Babel program are used to train a BLSTM based acoustic model. We consider a similar approach in this paper, where a single multilingual TDNN acoustic model is trained for all the four languages. A 5-layer TDNN is trained with a block-softmax output, with one output block for each of the languages. During training of the network, each mini-batch is balanced to contain examples from all languages. We use parallel training proposed in [21] for this purpose. The initial alignments for training the multilingual model are obtained from monolingual HMM/GMM acoustic models.

3. MULTILINGUAL LANGUAGE MODEL

In this section, we describe the architecture of our proposed multilingual neural language model, depicted in Figure 1, and explain its training procedure.

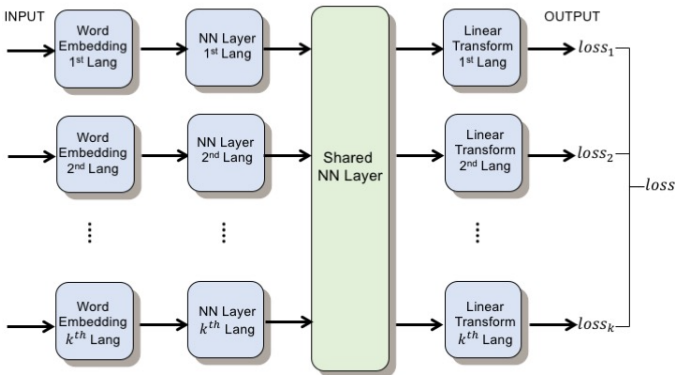

Fig. 1. Schematic of the proposed multilingual neural language model architecture with stacked layers. The elements in blue are language-specific and the green ones are language-independent. NN indicates a neural network model which is either TDNN or LSTM in our study.

As shown in Figure 1, for a given language i, first a language-specific word embedding maps the input batch of the given language to its embedding vectors. A language dependent neural network layer (either TDNN or LSTM) then captures the language-specific characteristics of the input, followed by a language-independent neural network layer (again either TDNN or LSTM), where its parameters are shared across all languages. In the next step, a language-specific linear transfer, followed by the softmax function, provides the predicted probability distribution $\hat{y}_i^{(t)}$ at timestep t over the vocabulary of the given language l.

For training, we use a joint multilingual objective that facilitates the sharing of a subset of parameters for each language $\theta_1, \ldots, \theta_M$ of our stacked neural language network as in [5]:

$$\mathcal{L}(\theta_1, \ldots, \theta_M) = -\frac{1}{Z} \sum_{t}^{N_e} \gamma_l \sum_{l}^{M} \mathcal{H}(y_t^{(l)}, \hat{y}_t^{(l)}) \qquad (1)$$

where $Z = M \times N_e$, N_e is the epoch size, γ_l is a hyper-parameter for each language objective which encodes prior knowledge about its importance and $\mathcal{H}$ is the cross-entropy loss between the ground-truth words and the predicted ones. Note that the sentence order in each language is preserved above and that the overall loss is back-propagated through the network, updating both language-specific and language-independent parameters. The sentences are processed in a cyclic fashion for the languages which have lesser number of sentences. Once the last sentence of the text corpus is processed for that language, the next sentence that is processed is the beginning one. The joint objective $\mathcal{L}$ can be minimized with respect to the parameters $\theta_1, \ldots, \theta_M$ using Stochastic Gradient Descent (SGD). This training strategy has been show beneficial in the past for multilingual document classification [5] and multilingual neural machine translation [22].

Table 1. *Statistics of the four BABEL languages.*

Statistics ↓	Languages			
	Tagalog	**Swahili**	**Turkish**	**Zulu**
no. of sentences (train)	93131	39354	82253	54660
no. of words (train)	594854	250398	573323	369476
no. of sentences (dev)	11191	9678	10297	9163
no. of words (dev)	73143	62875	73306	58285
vocab. size	22907	23956	41196	56885

4. EXPERIMENTS AND RESULTS

The experiments are performed on four babel languages, released as a part of the IARPA Babel program, namely Tagalog, Swahili, Turkish and Zulu. The detailed statistics of each language set are mentioned in Table 1. Each language has a held out development set, used for reporting the perplexity of language models as well as Word Error Rate (WER) of the ASR systems.

SRILM toolkit is used to create N-gram language models. The neural language models are implemented in *pytorch*. A modified version of TDNN, released in [13] is used in our experiments. We also implement a multilingual neural language model with RNN, following the work in [1]. The acoustic models are trained using the Kaldi speech recognition toolkit [23].

The full vocabulary size of all languages for the neural network training is used in this work. The neural networks are optimized using the SGD algorithm with early stopping, and negative log likelihood as the loss function. One-hot encoding is used for the input layers with the size of the number of vocabularies in each language. The dimensions of the hidden nodes of TDNN, LSTM, and RNN as well as the word embedding are set to 600 dimensions. The feed-forward architecture of the TDNN nodes consist of three hidden layers. All hyper-parameters of the model are set according to the best results, evaluated on the development set. The parameter γ_l in Equation 1 is tuned specifically for the best performance overall in this work, but it was not explored exhaustively for the range of values.

4.1. Language Modeling Results

As a baseline, we first compute the perplexity using N-gram language models. The N-gram used in this work is a tri-gram. A four-gram was also experimented but it was performing worse than the tri-gram. We test different variations of TDNN and LSTM networks, e.g. TDNN-LSTM refers to a model where the first neural network layer in Figure 1 (language-dependent) is a TDNN and the shared network is LSTM.

The perplexity of the monolingual N-grams for the dif-

Table 2. *Perplexity on the Babel Set of four languages for monolingual and multilingual LMs are presented. For the stacked models, the first network is language-dependent and second one is shared.*

LM ↓	Perplexity			
	Tagalog	**Swahili**	**Turkish**	**Zulu**
n-gram	148.1	357.5	396.5	719.9
RNN (multi) [1]	142.9	294.9	**284.3**	**665.0**
TDNN-LSTM (mono)	136.0	383.4	422.2	725.1
LSTM-LSTM (multi)	159.2	302.4	509.5	1550.0
TDNN-TDNN (multi)	137.4	622.2	293.9	1485.4
TDNN-LSTM (multi)	**133.9**	**284.5**	337.1	1006.0

Table 3. *WER on the Babel Set of four languages for N-gram, monolingual and multilingual LMs. Our model improves significantly WER on two out of four languages without hurting significantly the performance of the other two.*

LM ↓	WER %			
	Tagalog	**Swahili**	**Turkish**	**Zulu**
n-gram (baseline)	44.5	35.4	**46.1**	**54.2**
RNN (multi) [1]	44.7	35.3	46.3	55.4
TDNN-LSTM (mono)	43.8	35.3	46.7	55.0
LSTM-LSTM (multi)	44.8	35.7	47.2	55.9
TDNN-TDNN (multi)	44.4	35.5	46.3	55.8
TDNN-LSTM (multi)	**43.6**	**35.0**	46.5	55.3

ferent languages are shown in Table 2. It can be seen that Tagalog has good perplexity while Swahili, Turkish and Zulu have higher perplexities. This is due to the difference in the training data and the vocabulary size as seen in table 1. The multilingual TDNN-LSTM LM is seen to be performing well in terms of perplexity compared to the N-grams and also better than the monolingual TDNN-LSTM LM. Improvements can be seen for Tagalog, Swahili and Turkish although degradation is observed for Zulu.

The multilingual TDNN-LSTM LM is also compared to the TDNN-TDNN and LSTM-LSTM LM. The LSTM-LSTM LM suffers degradation in perplexity compared to the TDNN-LSTM LM as seen in table 2. The TDNN-LSTM LM is also better than the TDNN-TDNN LM except for Turkish. Note that the perplexity values in this work are different from the work in [1], since a full vocabulary size is used in our work. The results using the RNN LM system proposed in [1] are also computed and the TDNN-LSTM system is better except for Zulu and Turkish. The TDNN-LSTM LMs display good perplexity behavior in two of the four languages and results indicate that they perform well on languages with a small vocabulary.

In the following section, the proposed models are tested in terms of the word error rate for automatic speech recognition to see if the improvements in the perplexity actually result in WER improvement.

4.2. ASR Results

The N-gram language models are initially used for creating a graph for ASR decoding. Multilingual acoustic models are used and the setup has been described in section 2.2. The results using the N-gram decoding is shown in Table 3. The proposed neural networks are then used to re-score the N-best hypothesis generated from the lattices constructed using n-gram with weights (0.75 for the neural network and 0.25 for the n-gram) [24].

It can be seen that the WER for the Multilingual TDNN-LSTM system is better than the N-gram system and also better than the monolingual TDNN-LSTM system on Tagalog and Swahili languages. Comparisons are made to the other multilingual systems and it can be seen that the TDNN-LSTM system is better. The TDNN-LSTM multilingual system performs poorly for Zulu and Turkish. Zulu and Turkish are languages with high fluctuations in their morphological structures, causing high number of overall vocabularies in comparison to the others. In such cases, more simpler models like n-grams with lower parameters appear to be more effective, as they are less vulnerable to over-fitting. The WERs for Swahili are better than the ones reported in [1] even without applying the fine-tuning. The relative improvement in WER for Tagalog is 2 % and for Swahili is 1 % with respect to the N-grams.

5. CONCLUSION

This work examines the use of multilingual language models using neural architectures. A stacked TDNN-LSTM architecture is used where the TDNN models the long context and the LSTM models the sentence structure. Training the multilingual LMs involves adding the losses of each language and the total loss is back-propagated. Experiments show that the multilingual TDNN-LSTM architecture outperforms N-grams and other stacked neural architectures on two out of four languages in terms of both perplexity and word error rate. In the future, more languages will be used for training, while adaptation to a particular unseen language can also be performed as done in [1], to further improve the perplexity and word error rate. In this work, the weights of the multilingual LMs have not been explored exhaustively for the range of values. In the future, the weights can be tuned more effectively to get better performances.

6. REFERENCES

[1] Anton Ragni, Edgar Dakin, Xie Chen, Mark JF Gales, and Kate M Knill, "Multi-language neural network language models.," in *Interspeech*, 2016, pp. 3042–3046.

[2] Sibo Tong, Philip N Garner, and Hervé Bourlard, "An investigation of deep neural networks for multilingual speech recognition training and adaptation," in *Interspeech*, 2017.

[3] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *SLT Workshop*. IEEE, 2012, pp. 336–341.

[4] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7319–7323.

[5] Nikolaos Pappas and Andrei Popescu-Belis, "Multilingual hierarchical attention networks for document classification," in *Proc. IJNLP*, 2017.

[6] Melvin Johnson et al., "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association of Computational Linguistics*, 2017.

[7] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010.

[8] Stefan Kombrink, Tomáš Mikolov, Martin Karafiát, and Lukáš Burget, "Recurrent neural network based language modeling in meeting recognition," in *Interspeech*, 2011.

[9] Xunying Liu, Yongqiang Wang, Xie Chen, Mark JF Gales, and Philip C Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *ICASSP*. IEEE, 2014, pp. 4908–4912.

[10] Hainan Xu, Ke Li, Yiming Wang, Jian Wang, Shiyin Kang, Xie Chen, Daniel Povey, and Sanjeev Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *ICASSP*. IEEE, 2018.

[11] Gbor Melis, Chris Dyer, and Phil Blunsom, "On the state of the art of evaluation in neural language models," in *Proc. ICLR*, 2018.

[12] Takashi Wada and Tomoharu Iwata, "Unsupervised cross-lingual word embedding by multilingual neural language models," *arXiv preprint arXiv:1809.02306*, 2018.

[13] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[14] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[15] John Miller and Moritz Hardt, "When recurrent models don't need to be recurrent," *arXiv preprint arXiv:1805.10369*, 2018.

[16] Stanley F Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[17] Anoop Deoras, Tomáš Mikolov, Stefan Kombrink, Martin Karafiát, and Sanjeev Khudanpur, "Variational approximation of long-span language models for lvcsr," in *ICASSP*. IEEE, 2011, pp. 5532–5535.

[18] Gwénolé Lecorvé and Petr Motlicek, "Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition," in *Interspeech*, 2012.

[19] Heike Adel, Katrin Kirchhoff, Ngoc Thang Vu, Dominic Telaar, and Tanja Schultz, "Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding," in *Interspeech*, 2014.

[20] Martin Karafiát, Murali Karthick Baskar, Pavel Matějka, Karel Veselỳ, František Grézl, and Jan Černocky, "Multilingual blstm and speaker-specific vector adaptation in 2016 but babel system," in *SLT Workshop*. IEEE, 2016, pp. 637–643.

[21] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[22] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proc. NAACL-HLT*, San Diego, CA, USA, June 2016, pp. 866–875.

[23] Daniel Povey et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[24] Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky, "Rnnlm-recurrent neural network language modeling toolkit," in *Proc. of the 2011 ASRU Workshop*, 2011, pp. 196–201.