
Proceedings Paper:
Ragni, A., Dakin, E., Chen, X. et al. (2016) Multi-language neural network language models. In: Interspeech 2016. Interspeech 2016, 08-12 Sep 2016, San Francisco, CA, USA. International Speech Communication Association (ISCA). ISSN: 1990-9772.

https://doi.org/10.21437/interspeech.2016-371

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Multi-Language Neural Network Language Models

Anton Ragni, Edgar Dakin, Xie Chen, Mark J. F. Gales, Kate M. Knill

Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ, UK
{ar527,ed408,xc257,mjfg,kate.knill}@eng.cam.ac.uk

Abstract

In recent years there has been considerable interest in neural network based language models. These models typically consist of vocabulary dependent input and output layers and one, or more, hidden layers. A standard problem with these networks is that large quantities of training data are needed to robustly estimate the model parameters. This poses a challenge when only limited data is available for the target language. One way to address this issue is to make use of overlapping vocabularies between related languages. However this is only applicable to a small set of languages, and the impact is expected to be limited for more general applications. This paper describes a general solution that allows data from any language to be used. Here, only the input and output layers are vocabulary dependent whilst hidden layers are shared, language independent. This multi-task training set-up allows the quantity of data available to train the hidden layers to be increased. This multi-language network can be used in a range of configurations, including as initialisation for previously unseen languages. As a proof of concept this paper examines multilingual recurrent neural network language models. Experiments are conducted using language packs released within the IARPA Babel program.

Index Terms: recurrent neural network, language model, data augmentation, multi-task learning

1. Introduction

In domains such as speech recognition [1], machine translation [2] and many others, statistical language models are used to assign a probability to a word sequence. The standard approach is to use n-gram models [3]. Neural network language models [4, 5] are a powerful alternative. These models offer several benefits, such as continuous space representation by feed-forward models [6, 4], which may offer better generalisation, and context representation by recurrent models [7, 5], which may offer improved long range dependency modelling. Together these benefits have resulted in significant improvements over n-gram models on medium and large data sets [4, 5, 8].

In resource constrained conditions these neural network models may face robustness issues [7]. There are a number of possible solutions to adopt. These can be divided into two groups: model complexity reduction and robust parameter estimation [9, 10]. The former includes class-based [11, 12] and out-of-shortlist [4, 13, 5, 14, 15, 16] approaches to reduce the number of output layer parameters, parameter tying across context words [4] to reduce the number of input layer parameters. The latter includes augmentation schemes [17, 18, 9, 10, 19, 20, 21, 22, 23, 24, 25, 26]. Model-based augmentation schemes [17, 18], such as language model interpolation [17], make use of additional models to improve estimates. A combined model is formed where individual, in-domain and out-of-domain, models are represented with costs reflecting their usefulness on held-out data. Data-based schemes instead make use of data to initialise [27], train [20, 23] or adapt [24] the models. All these schemes rely on additional models and data not being orthogonal. There are situations where this assumption does not hold. One example is multilingual data. Excluding code-switching and loan words [28], the intersect of multilingual data vocabularies may be an empty set. Such situations cannot be handled using current augmentation approaches.

This paper proposes a general solution for training neural network language models on multilingual data. The idea can be thought of as creating a single network for all languages where some parameters are language specific and the rest are language independent. This enables the network to be trained in the standard fashion on data simultaneously from multiple languages. As language independent parameters are trained on larger quantities of data, the network may be expected to generalise better. These language independent parameters may also be ported to unseen languages simply by changing language specific parameters from one set to another. This may provide a better than random initialisation in limited resource conditions.

The rest of this paper is organised as follows. Section 2 provides a short overview to two frequently used neural network language models. Section 3 describes a general solution for training these models on multilingual data. Section 4 illustrates the solution using data from a number of diverse languages. Finally, Section 5 concludes the paper.

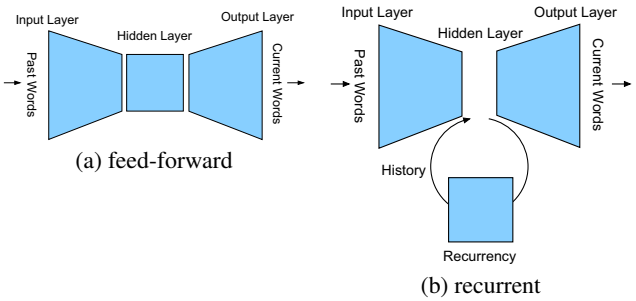
2. Neural network language models

Statistical language modelling using neural networks has seen a large amount of attention in domains such as speech recognition [29, 4, 5, 30, 31]. The number of possible approaches goes far beyond the scope of this paper. This section will focus on two forms: feed-forward [4] and recurrent [5].

A feed-forward neural network language model (FNN) [6, 4] is illustrated in Figure 1(a). This model uses a layered structure to yield a distribution over word vocabulary V given a fixed context of past words. Each context word is encoded using 1-of-$|V|$ encoding. These vectors then undergo a series of transformations passing through input, hidden, and output layer.

This work was supported 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. Disclaimer: 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 IARPA, DoD/ARL, or the U. S. Government.

http://dx.doi.org/10.21437/Interspeech.2016-371



(a) feed-forward

(b) recurrent

Figure 1: *Two neural network language models.*

The precise series of transformations can be described by

$$\mathbf{h}_t = \boldsymbol{\sigma}_{\mathtt{h}}(\mathbf{W}_{\mathtt{hh}} \cdot \boldsymbol{\sigma}_{\mathtt{i}}(\mathbf{W}_{\mathtt{hi}} \cdot \mathbf{x}_t)) \quad (1)$$

$$\mathbf{y}_t = \boldsymbol{\sigma}_{\mathtt{o}}(\mathbf{W}_{\mathtt{oh}} \cdot \mathbf{h}_t) \quad (2)$$

where $\mathbf{x}_t$, $\mathbf{h}_t$ and $\mathbf{y}_t$ are the context words prior to any transformations, after input and hidden layer transformations and after output layer transformations respectively. The last transformation yields the distribution over word vocabulary at time t. Transformations consist of a linear part represented by matrices $\mathbf{W}_{\mathtt{hi}}$, $\mathbf{W}_{\mathtt{hh}}$, $\mathbf{W}_{\mathtt{oh}}$ and non-linear part represented by functions $\boldsymbol{\sigma}_{\mathtt{i}}$, $\boldsymbol{\sigma}_{\mathtt{h}}$, $\boldsymbol{\sigma}_{\mathtt{o}}$. FNNs typically share input transformation across context words and use linear, hyperbolic tangent and soft-max functions as $\boldsymbol{\sigma}_{\mathtt{i}}$, $\boldsymbol{\sigma}_{\mathtt{h}}$ and $\boldsymbol{\sigma}_{\mathtt{o}}$ respectively.

A recurrent neural network language model (RNN) [5] is illustrated in Figure 1(b). Compared to the FNN, there is a feedback loop which passes information from the hidden layer back to the input layer. This information together with the current set of context words is then passed through the network

$$\mathbf{h}_t = \boldsymbol{\sigma}_{\mathtt{h}}(\mathbf{W}_{\mathtt{hi}} \cdot \mathbf{x}_t + \mathbf{W}_{\mathtt{hh}} \cdot \mathbf{h}_{t-1}) \quad (3)$$

$$\mathbf{y}_t = \boldsymbol{\sigma}_{\mathtt{o}}(\mathbf{W}_{\mathtt{oh}} \cdot \mathbf{h}_t) \quad (4)$$

where $\mathbf{h}_{t-1}$ is the hidden layer vector from the previous time, $\boldsymbol{\sigma}_{\mathtt{h}}$ and $\boldsymbol{\sigma}_{\mathtt{o}}$ are usually sigmoid and soft-max functions. Although a similar layered structure is used, the feedback loop enables to yield a distribution over word vocabulary in $\mathbf{y}_t$ given the complete past word history encapsulated in $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$.

FNNs and RNNs have three primary sets of parameters to estimate. These are the matrices $\mathbf{W}_{\mathtt{hi}}$, $\mathbf{W}_{\mathtt{hh}}$ and $\mathbf{W}_{\mathtt{oh}}$ associated with the input, hidden and output layer respectively. The input and output layer matrices by definition depend on vocabulary V whereas there is flexibility with the size $|H|$ of hidden layer matrix. The number of parameters is $\mathcal{O}(|V|^2)$ when $|H| \ll |V|$. Parameter estimation is typically performed by optimising the cross-entropy criterion over training data [4, 5].

3. Multi-language extension

The problem of training feed-forward neural network (FNN) models on multilingual data has received a lot of attention in acoustic modelling for speech recognition [32, 33, 34, 35]. In contrast to language modelling, the input to a FNN acoustic model typically consists of a fixed-dimensional feature vector extracted from raw audio using standard speech parametrisations [36, 37, 38, 34]. The output layer, similar to language modelling, is language dependent. The FNN acoustic model can be described using equations (1) and (2) where $\mathbf{x}_t$ is the feature vector at time t and $\mathbf{y}_t$ yields a distribution typically over sub-phonetic hidden Markov model (HMM) states. In order to train this network on multilingual data it is necessary to take into account language dependence in the output layer. One popular solution [32] is illustrated in Figure 2 for three languages: $L1$, $L2$ and $L3$. The solution can be thought of as creating a single network for all languages where the output layer parameters are language specific whereas the input and hidden layer parameters are language independent. Compared to Figure 1 (a), the output layer in Figure 2 consists of language specific blocks $\mathbf{W}_{\mathtt{oh}}^{(L1)}$, $\mathbf{W}_{\mathtt{oh}}^{(L2)}$ and $\mathbf{W}_{\mathtt{oh}}^{(L3)}$. Each block is used only with features extracted from the corresponding language. The precise series of transformation for the general case of L languages is

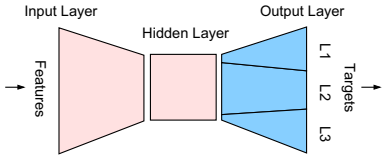


Figure 2: *Multi-language acoustic model.*

$$\mathbf{h}_t = \boldsymbol{\sigma}_{\mathtt{h}}(\mathbf{W}_{\mathtt{hh}} \cdot \boldsymbol{\sigma}_{\mathtt{i}}(\mathbf{W}_{\mathtt{hi}} \cdot \mathbf{x}_t)) \quad (5)$$

$$\mathbf{y}_t^{(l)} = \boldsymbol{\sigma}_{\mathtt{o}}\left(\mathbf{W}_{\mathtt{oh}}^{(l)} \cdot \mathbf{h}_t\right) \quad (6)$$

where $\mathbf{W}_{\mathtt{oh}}^{(l)}$ and $\mathbf{y}_t^{(l)}$ are the output layer parameters and the distribution over HMM states for language $l \in [1, L]$. Such solution enables the quantity of data available to train the input and hidden layers to be increased which has been shown to improve robustness in limited resource conditions [35].

In neural network language models discussed in Section 2 both input and output layers are language-dependent. If the same approach was applied to the input layer it then would have been possible to train these models on multilingual data using standard approaches. Figure 3 illustrates such a solution for RNN language models. Compared to Figure 1 (b), the input and output layers in Figure 3 consist of language specific blocks $\mathbf{W}_{\mathtt{hi}}^{(L1)}$, $\mathbf{W}_{\mathtt{hi}}^{(L2)}$, $\mathbf{W}_{\mathtt{hi}}^{(L3)}$ and $\mathbf{W}_{\mathtt{oh}}^{(L1)}$, $\mathbf{W}_{\mathtt{oh}}^{(L2)}$, $\mathbf{W}_{\mathtt{oh}}^{(L3)}$ respectively. A matching pair of language specific blocks, such as $\mathbf{W}_{\mathtt{hi}}^{(L2)}$ and $\mathbf{W}_{\mathtt{oh}}^{(L2)}$ is used only with the corresponding language ($L2$). For the general case of L languages

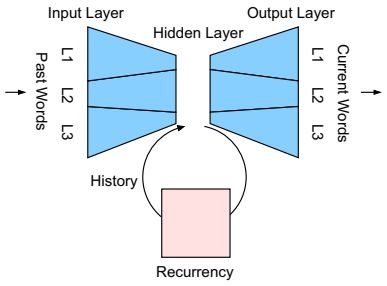


Figure 3: *Multi-language language model.*

$$\mathbf{h}_t = \boldsymbol{\sigma}_{\mathtt{h}}(\mathbf{W}_{\mathtt{hi}}^{(l)} \cdot \mathbf{x}_t^{(l)} + \mathbf{W}_{\mathtt{hh}} \cdot \mathbf{h}_{t-1}) \quad (7)$$

$$\mathbf{y}_t^{(l)} = \boldsymbol{\sigma}_{\mathtt{o}}(\mathbf{W}_{\mathtt{oh}}^{(l)} \cdot \mathbf{h}_t) \quad (8)$$

where $\mathbf{W}_{\mathtt{hi}}^{(l)}$, $\mathbf{W}_{\mathtt{oh}}^{(l)}$ are language specific parameters trained on language l data and $\mathbf{W}_{\mathtt{hh}}$ are language independent parameters trained on all data. It may be expected that increased quantities of data available to hidden layers should enhance generalisation.

Once the multi-language model is available it can be fine-tuned for each individual language. Another interesting option is to port language independent layers to initialise models for unseen languages. This can be illustrated for language $L+1$ by

$$\mathbf{h}_t = \boldsymbol{\sigma}_{\mathtt{h}}(\mathbf{W}_{\mathtt{hi}}^{(L+1)} \cdot \mathbf{x}_t^{(L+1)} + \mathbf{W}_{\mathtt{hh}} \cdot \mathbf{h}_{t-1}) \quad (9)$$

$$\mathbf{y}_t^{(L+1)} = \boldsymbol{\sigma}_{\mathtt{o}}(\mathbf{W}_{\mathtt{oh}}^{(L+1)} \cdot \mathbf{h}_t) \quad (10)$$

where $\mathbf{W}_{\mathtt{hh}}$ are hidden layer parameters of the multi-language model and $\mathbf{W}_{\mathtt{hi}}^{(L+1)}$, $\mathbf{W}_{\mathtt{oh}}^{(L+1)}$ are language $L+1$ specific input and output layer parameters. The hidden layer parameters may initially be locked to the multi-language estimate by setting the hidden layer learning rate to zero. This enables to take advantage of robustly estimated language independent hidden layer parameters in training the input and output layers. Once these have been trained the hidden layer can be unlocked and training continued. This will be referred to as fine-tuning in this paper and it can be done for both multi-language training and porting.

4. Experiments

As a proof of concept this paper considers training recurrent neural network language models on multilingual data. The multilingual data for this work is taken from language packs released within IARPA Babel program [39].[1] These packs contain up to 60 hours of conversational telephone speech data. Each language additionally has 10 hours of held-out data for development. A total of 14 languages are considered in this paper. These languages come from different geographic locations and represent a diverse set of families. Table 1 shows that there is a wide variety of vocabulary sizes starting from Vietnamese with 6,685 words and ending with Zulu with 60,916 words. Out-of-vocabulary rate on held-out data for these languages varies from near zero for Vietnamese (syllabic) to 11% for Zulu.

Table 1: *Summary of languages and amounts of training data.*

Families	Languages	Words	
		Unique	Train
Austronesian	Tagalog	23,705	610,642
Austronesian	Cebuano	15,737	346,986
Austro-Asiatic	Vietnamese	6,685	962,311
Dravidian	Tamil	57,799	462,321
Indo-European	Assamese	25,455	443,093
Indo-European	Bengali	28,847	517,128
Indo-European	Pashto	21,051	982,589
Turkic	Turkish	45,513	612,788
Tai-Kadai	Lao	7,942	628,876
Sino-Tibetan	Cantonese	23,964	884,697
Niger-Congo	Zulu	60,916	543,587
Niger-Congo	Swahili	24,361	394,578
Nilo-Saharan	Dholuo	17,550	467,206
Creole	Haitian Creole	14,812	622,489

[1]Language identifiers: Cantonese IARPA-babel101b-v0.4c, Assamese IARPA-babel102b-v0.5a, Bengali IARPA-babel103b-v0.4b, Pashto IARPA-babel104b-v0.4aY, Turkish IARPA-babel105b-v0.4, Tagalog IARPA-babel106-v0.2f, Vietnamese IARPA-babel107b-v0.7, Haitian Creole IARPA-babel201b-v0.2b, Lao IARPA-babel203b-v3.1a, Tamil IARPA-babel204b-v1.1b, Zulu IARPA-babel206b-v0.1d, Cebuano IARPA-babel301b-v2.0b, Swahili IARPA-babel202b-v1.0d, Dholuo IARPA-babel403b-v1.0b.

Experiments were conducted using CUED RNN language modelling toolkit [40] extended to support the multi-language language models depicted in Figure 3. Unless otherwise stated weights $\{\mathbf{W}_{\mathtt{hi}}^{(l)}\}$, $\mathbf{W}_{\mathtt{hh}}$ and $\{\mathbf{W}_{\mathtt{oh}}^{(l)}\}$ in equations (7) and (8) were randomly initialised. Training is performed by optimising the cross-entropy criterion using back propagation through time with the time step set to 5 [41]. Training sentences were randomised and organised into spliced bunches of 64 sentences. Word vocabularies in the output layers were down sampled to 75% of the full vocabulary sizes. The remaining 25% of the least likely according to unigram statistics words were represented using an out-of-shortlist (OOS) word node. The hidden layer weight $\mathbf{W}_{\mathtt{hh}}$ in this work is a 100×100 matrix.

A preliminary experiment was conducted among three geographically linked African continent languages – Dholuo, Swahili and Zulu – with the last two languages belonging to the same family. It should be noted that Swahili unlike Zulu is a trading language that exhibits lots of borrowing from Arabic and European languages. Three language specific and one multi-language RNNs were trained. The amount of training data for the multi-language RNN is 1.4 million words and approximately third of that is available to each language specific RNN. Perplexities for the training languages on their respective development sets are shown in Table 2. The multi-language RNN (`multi`) shows perplexity reductions over language specific RNNs (`mono`) on all languages. The largest gain comes for Zulu where perplexity is reduced relatively by 8%. Overall gain is rather moderate despite two thirds of the data being from the same language family. The average perplexity across three languages drops from 538.2 to 506.5. Although average perplexities do not account for the difference in vocabulary sizes, a simple metric such as average is useful to assess general trends.

Table 2: *Perplexities for language specific and multi-language RNNs for selected African continent languages.*

Languages	Structure	
	`mono`	`multi`
Dholuo	203.9	198.3
Swahili	358.0	349.0
Zulu	1052.7	972.3
Average	538.2	506.5

The use of multilingual data has so far considered related languages. In order to assess the impact of using less related languages, a set of 11 mixed-origin languages from Table 1 was chosen. The amount of training data for the multi-language RNN in this case is 7.3 million words. Perplexities in this experiment are shown in the first two columns of Table 3. Overall, the multi-language RNN yields lower perplexities than language specific RNNs though there are languages where perplexity is a bit higher. The average drop in perplexity is smaller than that obtained across related languages in Table 2. Comparing the use of related and mixed-origin languages for the only language occurring in both tables, Zulu, it can be seen that using data from related languages is a little bit more advantageous.

The multi-language RNNs in Table 2 or 3 provide a model optimal for all training languages. This may not be optimal for each individual language and hence may impact possible gains. A simple way to verify this is to fine-tune the multi-language RNN for each language. As discussed in Section 3, there are options how fine-tuning can be performed. Initially, only input and output layers were fine-tuned whilst the hidden layer was

Table 3: *Perplexities for language specific and multi-language RNNs for 11 mixed-origin languages.*

Languages	Structure		Fine tuning	
	mono	multi	$\mathbf{W}_{\mathtt{hi}}, \mathbf{W}_{\mathtt{oh}}$	$+\mathbf{W}_{\mathtt{hh}}$
Tagalog	135.7	136.7	133.6	129.2
Assamese	321.3	318.8	304.0	297.6
Bengali	355.5	358.1	342.8	333.4
Creole	137.2	138.9	135.7	131.4
Lao	105.1	107.8	103.7	100.8
Tamil	925.5	903.4	866.6	851.7
Zulu	1052.7	987.9	982.7	984.7
Cantonese	123.3	122.1	119.4	116.6
Pashto	147.0	148.1	143.2	138.3
Turkish	443.8	434.1	418.3	405.9
Vietnamese	136.6	138.8	132.4	127.2
Average	353.1	345.0	334.8	328.8

locked to the estimate obtained from 11 languages. This allows input and output layer for each language to be tuned for the final language independent recurrent layer. Perplexities for this experiment are shown in the third column of Table 3. These initial RNNs with the multi-language hidden layer show perplexity reductions over language specific RNNs (`mono`) for all languages. Training was then continued with the hidden layer unlocked to adjust all parameters to the target language. As shown by the last column in Table 3, further perplexity reductions can be obtained for all language apart from Zulu.

Experiments so far have examined only seen languages. An important question is whether the language-independent multi-language RNN parameters can generalise to unseen languages. This can be measured by porting hidden layers to held-out language RNNs. A particular interest is whether this can help in resource constrained conditions. For such experiments very limited language packs (VLLP) containing 3 hours of conversational telephone speech were used. Two held-out languages, Cebuano and Swahili, were chosen for evaluation. The amount of training data in VLLP conditions is 31,959 and 24,703 words respectively. This is more than 10 times less than in FLP conditions. The number of unique words is 3,614 and 5,475 respectively. Perplexities for this experiment are summarised in Table 4. Language specific RNNs with randomly (`random`) initialised hidden and input and output layers are shown on the first line. The second line shows perplexities of language specific RNNs with the imported language independent hidden layer and randomly initialised input and output layers. The same multi-stage fine-tuning process was perfored as in Table 3. These results suggest that the language independent multi-language hidden layer encapsulates information useful for perplexity reduction. Overall gains for Cebuano are quite small and for Swahili are moderate 7% relative reductions.

Table 4: *Perplexities for language specific RNNs with imported language independent multi-language hidden layer*

Layer Initialisation		Language	
$\mathbf{W}_{\mathtt{hh}}$	$\mathbf{W}_{\mathtt{hi}}, \mathbf{W}_{\mathtt{oh}}$	Cebuano	Swahili
random	random	169.6	532.7
multi	random	164.5	496.0

The final set of experiments examined whether perplexity reductions seen in Table 4 will translate into word error rate (WER) reductions when these language models are used for speech recognition. Acoustic models in these experiments are based on multilingual bottleneck features [42, 34, 32] extracted from a feed-forward neural network (FNN) trained on the FLP audio data of the same 11 languages shown in Table 3. Training of such multi-language acoustic models was discussed in Section 3. These bottleneck features provide useful additional information and are crucial for accurate speech transcription in resource constrained conditions [35]. Two types of acoustic models are trained for each language: a Tandem and Hybrid [43]. These acoustic models differ in the form of final classifier: Gaussian mixture models (Tandem) and FNN (Hybrid). In order to take advantage of system combination benefits, these acoustic models are combined in a multi-stream fashion during inference using joint decoding [43]. Speech recognition results for Cebuano and Swahili VLLP are shown in Table 5. Each block of results shows performances of trigram (`n-gram`), language-specific (`mono-rnn`) and ported multi-language RNN (`multi-rnn`) (line 2 in Table 4) language model respectively. The RNNs were interpolated with the n-gram model using equal costs ($\frac{1}{2}, \frac{1}{2}$). Language specific RNNs compared to trigrams show mixed performance with gain seen for Cebuano and no gain seen for Swahili. The latter performance is in line with expectations whereas the former is a bit surprising given limited amount of training data. The multi-language RNN shows small improvements for both languages.

Table 5: *Cebuano and Swahili VLLP word error rates (WER) with n-gram, language specific RNN and multi-language RNNs.*

Language	LM	WER (%)
Cebuano	n-gram	62.1
	mono-rnn	61.3
	multi-rnn	61.2
Swahili	n-gram	56.3
	mono-rnn	56.3
	multi-rnn	56.2

5. Conclusions

Training accurate statistical language models, such as recurrent neural networks, on small amounts of data is a challenging. Often it is possible to acquire additional data which may originate from a different source. If there is virtually no overlap between data sources then standard approaches, such as language model interpolation and data augmentation, cannot be used. One example is multilingual data. This paper proposed a general solution suitable for incorporating multilingual data into training of two neural network language models: feed-forward and recurrent. The solution is based on the observation that typically only input and output layers of such models depend on vocabulary. Hence, a neural network with shared hidden layers and language dependent input and output layers can be trained on multilingual data using standard approaches. As a proof of concept this paper presented recurrent neural network language model training on multilingual data. A total of 14 diverse languages provided by the IARPA Babel program were considered. Results suggest that shared hidden layer representations can help to reduce perplexity of individual languages. Furthermore, such representations can generalise to unseen languages.

6. References

[1] F. Jelinek, *Statistical methods for speech recognition*. MIT Press, 1998.

[2] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

[3] J. T. Goodman, "A bit of progress in language modeling. extended version," Microsoft Research, Tech. Rep. MSR-TR-2001-72, 2001.

[4] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, 2007.

[5] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010, pp. 1045–1048.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[7] J. L. Elman, "Learning and development in neural networks: the importance of starting small," *Cognition*, vol. 48, pp. 71–99, 1993.

[8] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *ICASSP*, 2015, pp. 5391–5395.

[9] R. Iyer, M. Ostendorf, and H. Gish, "Using out-of-domain data to improve in-domain language models," *IEEE Signal Processing Letters*, vol. 4, no. 8, pp. 221–223, 1997.

[10] R. Iyer and M. Ostendorf, "Transforming out-of-domain estimates to improve in-domain language models," in *Eurospeech*, 1997, pp. 2635–2639.

[11] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *AIS*, 2005, pp. 246–252.

[12] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," in *Interspeech*, 2010.

[13] A. Emami and L. Mangu, "Empirical study of neural network language models for arabic speech recognition," in *ASRU*, 2007.

[14] J. Park, X. Liu, M. J. F. Gales, and P. C. Woodland, "Improved neural network based language modelling and adaptation," in *Interspeech*, 2010.

[15] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *ICASSP*, 2011, pp. 5524–5527.

[16] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *ICASSP*, 2014.

[17] F. Jelinek and R. L. Mercer, "Interpolated estimation of markov source parameters from sparse data," in *Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.

[18] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *ICASSP*, 2003.

[19] R. Iyer and M. Ostendorf, "Relevance weighting for combining multi-domain data for n-gram language models," *Computer Speech and Language*, vol. 13, pp. 267–282, 1999.

[20] H. Schwenk and J.-L. Gauvain, "Training neural network language models on very large corpora," in *HLT/EMNLP*, 2005, pp. 201–208.

[21] R. Gretter and G. Riccardi, "On-line learning of language models with word error probability distributions," in *ICASSP*, 2001, pp. 557–560.

[22] A. Stolcke, "Error modeling and unsupervised language modeling," in *NIST LVCSR*, 2001.

[23] S. Novotney, R. Schwartz, and J. Ma, "Unsupervised acoustic and language model training with small amounts of labelled data," in *ICASSP*, 2009, pp. 4297–4300.

[24] M. Mahajan, D. Beeferman, and X. D. Huang, "Improved topic-dependent language modelling using information retrieval techniques," in *ICASSP*, 1999.

[25] X. Zhu and R. Rosenfield, "Improving trigram language modelling with the world-wide-web," in *ICASSP*, 2001.

[26] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *HLT*, 2003.

[27] Y. Shi, M. Larson, and C. M. Jonker, "Recurrent neural network language model adaptation with curriculum learning," *Computer Speech and Languages*, vol. 33, no. 1, pp. 136–154, 2015.

[28] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, "Recurrent neural network language modeling for code switching conversational speech," in *ICASSP*, 2013.

[29] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE TSP*, vol. 45, no. 11, pp. 2673–2681, 1997.

[30] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Interspeech*, 2012.

[31] E. Arisoy, A. Sethy, B. Ramabhadran, and S. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *ICASSP*, 2015.

[32] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012, pp. 336–341.

[33] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013, pp. 7319–7323.

[34] Z. Tüske, D. Nolden, R. Schluter, and H. Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014.

[35] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nussbaum-Thom, M. Picheny, Z. Tüske, P. Golik, R. Schlüter, H. Ney, M. Gales, K. Knill, A. Ragni, H. Wang, and P. Woodland, "Multilingual representations for low resource speech recognition and keyword search," in *ASRU*, 2015.

[36] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Speech and Audio Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[37] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoustic Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[38] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Willams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in *ICASSP*, 2013, pp. 8604–8608.

[39] M. Gales, K. Knill, A. Ragni, and S. Rath, "Speech recognition and keyword spotting for low-resource languages: BABEL project research at CUED," in *SLTU*, 2014.

[40] X. Chen, X. Liu, Y. Qian, M. J. F. Gales, and P. C. Woodland, "CUED-RNNLM – an open-source toolkit for efficient training and evaluation of recurrent neural network language models," in *ICASSP*, 2016.

[41] T. Mikolov, L. Kombrink, Burget, J. Černocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *ICASSP*, 2011.

[42] Z. Tüske, J. Pinto, D. Wilett, and R. Schlüter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *ICASSP*, 2013, pp. 7349–7353.

[43] H. Wang, A. Ragni, M. J. F. Gales, K. M. Knill, P. C. Woodland, and C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Interspeech*, 2015.