LDA-BASED CONTEXT DEPENDENT RECURRENT NEURAL NETWORK LANGUAGE MODEL USING DOCUMENT-BASED TOPIC DISTRIBUTION OF WORDS

Md. Akmal Haidar and Mikko Kurimo

Department of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Finland
akmalcuet00@yahoo.com, mikko.kurimo@aalto.fi

ABSTRACT

Adding context information into recurrent neural network language models (RNNLMs) have been investigated recently to improve the effectiveness of learning RNNLM. Conventionally, a fast approximate topic representation for a block of words was proposed by using corpus-based topic distribution of word incorporating latent Dirichlet allocation (LDA) model. It is then updated for each subsequent word using an exponential decay. However, words could represent different topics in different documents. In this paper, we form document-based distribution over topics for each word using LDA model and apply it in the computation of fast approximate exponentially decaying features. We have shown experimental results on a well known Penn Treebank corpus and found that our approach outperforms the conventional LDA-based context RNNLM approach. Moreover, we carried out speech recognition experiments on Wall Street Journal corpus and achieved word error rate (WER) improvements over the other approach.

***Index Terms*—** Recurrent neural network, language modeling, latent Dirichlet allocation, speech recognition

1. INTRODUCTION

Statistical n-gram language models (LMs) is an important part for many applications such as speech recognition, information retrieval, machine translations, etc. They generalize poorly due to insufficiencies of training data which encounters a data sparseness problem and it is traditionally handled by using backoff smoothing approaches with lower-order language models [1, 2]. Moreover, they cannot capture the long-range information of natural language. Several approaches such as cache-based LM [3], topic models [4, 5, 6] have been used to capture long-range information of natural language [7, 8, 9]. Recently, recurrent neural network LM (RNNLM) [10] have been shown a big impact in the language modeling research. These models are different from a classical feed-forward neural network language model (FFNNLM) [11].

A FFNNLM avoids the data sparseness problem by learning distributed representation of words as non-linear combinations of weights in a neural net. Here, the recent history with a fixed length window are mapped into a continuous space and then the word probabilities given the history words are estimated. In a FFNNLM, long-term dependency can only be captured with increasing computational cost in linear way. Recurrent NNLM (RNNLM) [10] can capture long-term dependencies by using a recurrent connection in the hidden layer from the previous time step. The long-range context information are stored in the hidden layer as a memory of the model. However, the RNN is theoretically powerful and is considered hard to train practically because of the so-called vanishing and exploding gradient problems [12, 13]. A simple efficient strategy of gradient clipping was introduced in [14] to avoid the exploding gradient problem.

Nevertheless, the RNN suffers from the gradient vanishing problem as the gradient backpropagated in time and their magnitude shrink close to zero. As a result, it is difficult for the model to learn longer terms [15]. Many methods have been proposed to overcome this problem. A complex model named as long short term memory RNN (LSTM-RNN) LM [16] was investigated where the recurrent hidden units are replaced with LSTM cell incorporating gating units. In [15], a modification of the RNNLM was introduced where the context information is learned in a context layer. Various kinds of pre-trained features have been incorporated into RNNLM [17, 18, 19, 20, 21].

In [17], an LDA-based RNNLM defined as (RNN-LDA) LM was proposed where topic distribution of a block of preceding words of the current word are computed. However, an LDA representation for each word given its sentence prefix is an expensive process. To avoid this process, an efficient fast approximate topic representation was investigated where the topic distribution for a block of words is computed by renormalizing the multiplication of individual distribution over topics for each word in the block and updated for each subsequent word using an exponential decay. The topic distribution for each word is created by normalizing the word probabilities for topics of the corpus derived using the LDA model. We define it in this paper as corpus-based topic distribution of word. However, words may describe different topics in different documents. For example, the word $bank$ may describe financial topic in one document and river topic in another document [22]. Therefore, document-based topic distribution of words should be more appropriate in the computation of the above approximate features. This motivates us to modify the RNN-LDA LM using document-based topic distribution of words and we define our approach as document RNN-LDA (DRNN-LDA) LM.

2. LATENT DIRICHLET ALLOCATION

Latent Dirichlet allocation (LDA) is a three-level hierarchical Bayesian model. It is a generative probabilistic topic model for documents in a corpus. The documents are represented by random latent topics, which are characterized by a distribution over words. The LDA model can be described in the following way. Each document $d = w_1, \cdots, w_N$ is generated as a mixture of unigram models, where the topic mixture vector θ_d is drawn from the Dirichlet distribution with parameter α. Here, N is the document length sampled from a Poisson distribution. The corresponding topic sequence $z = z_1, \cdots, z_N$ is generated using the multinomial distribution θ_d. Each word w_n is generated using the distribution $p(w_n|z_n, \beta)$. The Dirichlet priors α and β are the corpus level parameters that are assumed to be sampled once in generating the corpus. The parameters θ are document-level variables and sampled once per document. The variables z and w are word-level variables and sampled once for each word in each document. After LDA training, an inference method can be applied to obtain the topic distribution of an unseen document. A varional inference method for learning the model can be found in [6]. In this paper, we have used the implemention of [6] found in (http://www.cs.princeton.edu/∼blei/lda-c/) for LDA training and inference.

3. RECURRENT NEURAL NETWORK LM WITH FEATURE LAYER

RNNLM with feature layer [17] contains an input layer, a feature layer, a hidden layer and an output layer. The hidden layer has a recurrent connection that allows the propagation of the previous state information of the hidden layer. The feature layer is connected to both the hidden and output layer. The weight of each connection is stored in a weight matrix. In Figure 1, an input vector $w(t)$ encodes an input word at time t using 1-of-A encoding, also known as one-hot representation. It uses an index to each word in the vocabulary of size A and a word is encoded with 1 in its index position and all other coefficients are set to 0. The feature layer $k(t)$ contains the context information created using RNN-LDA or DRNN-LDA LMs. The output layer produces a probability distribution over words at time t given the information in the hidden layer and the feature layer. The output vector $y(t)$ also has the same dimension as the input vector $w(t)$. The hidden layer $h(t)$ stores the previous information and acts as a memory of the model. The values in the hidden and output layers are calculated as:

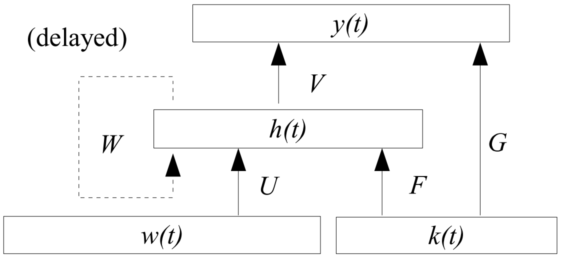

Fig. 1. Recurrent neural network with feature layer.

$$h(t) = f(Uw(t) + Wh(t-1) + Fk(t)) \quad (1)$$

$$y(t) = g(Vh(t) + Gk(t)) \quad (2)$$

where U, W, V, F, and G represent the input, recurrent, output, feature input and feature output weight matrices respectively. $f(z)$ and $g(z_m)$ are the sigmoid and the soft-max function respectively. The soft-max function in the output layer confirms that the output forms a valid probability distribution. To reduce the computation in the output layer, a simple hierarchy of two level soft-max approaches using frequency-based clustering was investigated in [23]. The training of the model is to learn the weight matrices that maximize the likelihood of the training data and it uses validation data for early stopping and to control learning rate [23]. The model is trained by using stochastic gradient descent with backpropagation through time ($BPTT$) algorithm [24]. Further details can be found in [17, 23].

3.1. Conventional and Proposed Fast Approximate Topic Representations

For RNN-LDA LM, a fast approximate topic representation for a block of first L words is formed by normalizing the multiplication of individual distribution over topics for each word in the block [17]:

$$k_{RNN-LDA}(t) = \frac{1}{Z}\prod_{i=0}^{L-1} p(z|w(t-i)), \quad (3)$$

and it is updated for each subsequent word by using an exponential decay [17]:

$$k_{RNN-LDA}(t) = \frac{1}{Z}k_{RNN-LDA}(t-1)^{\gamma}(p(z|w(t))^{(1-\gamma)}, \quad (4)$$

where γ is an exponential decaying parameter. The distribution $p(z|w(t))$ is a vector that describes the probabilities of topics given word $w(t)$. The vector is obtained by normalizing the distribution $p(w(t)|z, \beta)$ obtained using the LDA model [6]. Since the topics are equally represented in the training data, $p(z|w(t))$ can be obtained as:

$$p(z|w(t)) = \frac{p(w(t)|z,\beta)}{\sum_z p(w(t)|z,\beta)}. \quad (5)$$

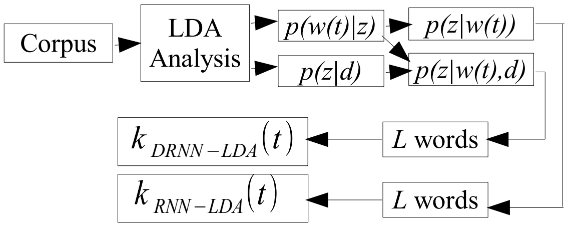

Fig. 2. Topic representation of L words using LDA.

In this paper, we replace the corpus-based topic distribution of word $p(z|w(t))$ used in [17] with document-based topic distribution of word which can be defined as:

$$p(z|w(t), d) = \frac{p(w(t)|z, \beta)p(z|d)}{\sum_z p(w(t)|z, \beta)p(z|d)}, \qquad (6)$$

where $p(z|d)$ is the vector that represents the topic distribution of document d. $p(z|d)$ can be obtained using LDA training and inference procedures [6]. The fast approximate topic representation for a block of first L words is then created for DRNN-LDA LM as:

$$k_{DRNN-LDA}(t) = \frac{1}{Z}\prod_{i=0}^{L-1} p(z|w(t-i), d), \qquad (7)$$

Then, we update the feature vector $k_{DRNN-LDA}(t)$ for each subsequent word using an exponential decay as [17]:

$$k_{DRNN-LDA}(t) = \frac{1}{Z}k_{DRNN-LDA}(t-1)^{\gamma}(p(z|w(t), d)^{(1-\gamma)}. \qquad (8)$$

It should be mention here that the constraint for this approximation to work is to add a small constant to smooth the distribution $p(w(t)|z, \beta)$ to avoid extremely small probabilities [17]. In this paper, we use the value of the constant as $1/A$, where A is the size of vocabulary. Figure 2 describes the generation of topic representation of L words for RNN-LDA and DRNN-LDA LMs.

4. EXPERIMENTS

4.1. Data and Experimental Details

We evaluated our approach using a well known Penn Treebank (PTB) corpus [17, 25] and Wall Street Journal (WSJ) corpus [26]. The PTB corpus was used for perplexity evaluation. For WER experiments, we selected one million (1 M) words from '87-89 WSJ text corpus (37 M words) and the transcription data (17 K words) of si_dt_20 folder from CSR-I (WSJ0) corpus [26] as the training and validation set respectively. The WER experiments are evaluated on the evaluation test, which is a total of 333 test utterances (5643 words) from the November 1992 ARPA CSR benchmark test data for non-verbalized vocabularies of 20K words [27]. In our experiments, we replaced the words that are not appeared in the vocabularies with a token <UNK>. The details of the corpora are described in Table 1. We used an RNNLM toolkit [28] to train the RNNLM and developed a modified version of the RNNLM toolkit to train the RNN-LDA LM and DRNN-LDA LMs. The SRILM toolkit [29] and the HTK toolkit [30] are used for generating the baseline LM and computing the word error rate (WER) respectively. The baseline model denoted as KN5 which is obtained by an interpolated 5-gram model with modified Kneser-Ney smoothing and no count cutoffs. For LDA training and inference, we use each non-overlapping sentence of a corpus as a document. We train and infer the LDA model for 40 topics and used a fixed α across topics. We obtained the topic distribution for training, validation and test documents by normalizing variational Dirichlet parameters found in LDA training and inference procedure [6]. We used a block size of L=50 words. Using PTB corpus, the best results for RNN-LDA and DRNN-LDA LMs are achieved using $\gamma = 0.95$ and $\gamma = 0.45$ respectively. For WSJ corpus, the values of $\gamma = 0.9$ and 0.4 give the best results for the RNN-LDA and DRNN-LDA LMs respectively. From Equation 4 and 8, we can note that the smaller value of γ indicates that the proposed document-based topic distribution of word generates more topical information than the corpus-based topic distribution of word. The acoustic model used for the WER experiments are taken from [31]. It is trained by using all WSJ and TIMIT training data, the 40-phone set of the CMU dictionary, approximately 10000 tied-states, 32 Gaussians per state and 64 Gaussians per silence state. The acoustic waveforms are parameterized into a 39-dimensional feature vector consisting of 12 cepstral coefficients plus the 0^{th} cepstral, delta and delta delta coefficients, normalized using cepstral mean subtraction ($MFCC_{0-D-A-Z}$). We evaluated the cross-word models. The values of the beam width, word insertion penalty, and the language model scale factor are 350.0, -4.0, and 15.0 respectively [31].

Corpus	**#Words**			**#VOC**
	Train	Valid	Test	
PTB	930 K	74 K	82 K	10 K
WSJ	1 M / 37 M	17 K	5643	17662

Table 1. Corpora

4.2. Result on PTB Corpus

In Table 2, we reported validation (Valid) and test perplexity (PPL) results on PTB corpus using various number of hidden neurons. Here, we used a factorization of the output layer using class size of 100 and 5 $BPTT$ steps [23]. From Table 2, we can note that the proposed DRNN-LDA LM approach outperforms the RNNLM and the RNN-LDA [17] LM approaches for all hidden neuron sizes. The best test perplexity result using DRNN-LDA LM is **114.0** which is obtained using 200 hidden neurons. We performed further experiments without factorization of the output layer using 200 hidden neurons and the test perplexity results are described in Table 3. Here, the result for RNN is reported from [23]. From Table 3, we can note that our proposed DRNN-LDA LM approach gives about 14.7% (123 to **104.9**) and 10.8% (117.6 to **104.9**) perplexity reduction over the RNNLM [23] and the conventional RNN-LDA LM approaches [17] respectively. The results in (**+KN5**) are obtained by interpolating the models with the baseline model using interpolation weight 0.5.

Language Model	H	**Valid PPL**	**Test PPL**
KN5	-	148.0	141.2
RNN	10	239.2	225.0
RNN-LDA	10	199.5	188.7
DRNN-LDA	10	**173.8**	**164.2**
RNN	100	150.1	142.1
RNN-LDA	100	136.5	130.7
DRNN-LDA	100	**122.0**	**115.7**
RNN	200	142.2	135.2
RNN-LDA	200	132.0	125.7
DRNN-LDA	200	**119.1**	**114.0**

Table 2. PPL results on PTB corpus using class size of 100, $BPTT$ = 5, and different number of hidden (H) neurons.

LM	**Individual**	**+KN5**
KN5	141.2	-
RNN	123	106
RNN-LDA	117.6	102.9
DRNN-LDA	**104.9**	**94.2**

Table 3. Test PPL results on PTB corpus using 200 hidden neurons, $BPTT$ = 5, and without factorizing the output layer.

4.3. Results on WSJ Corpus

We created lattices using pruned trigram with modified Kneser-Ney (KN) smoothing, from which we generated 100-best lists that are used for the rescoring experiments. The baseline n-gram language model for rescoring is a modified KN 5-gram (KN5) model with no count cutoff. The RNNLM, the RNN-LDA and the DRNN-LDA LMs are trained using 5 *BPTT* steps and using a class layer in the output layer. The models are interpolated with the baseline KN5 model with weight 0.75 for the RNNLM, the RNN-LDA or the DRNN-LDA LM and 0.25 for the baseline KN5 model. The evaluation test results on 1 M words of WSJ corpus using class size of 100 is described in Table 4. From Table 4, we can note that the interpolation of DRNN-LDA and KN5 LMs (DRNN-LDA+KN5) gives 11.7%(12.8% to **11.3**%), 4.2%(11.8% to **11.3**%), and 2.6%(11.6% to 11.3%) relative WER reductions over the KN5, the RNN+KN5, and the RNN-LDA+KN5 LMs respectively. We can also note that the RNN-LDA+KN5 LM yields 0.2% absolute WER improvements over RNN+KN5 LM as in [17] whereas the proposed DRNN-LDA+KN5 LM gives 0.5% absolute WER improvements with larger perplexity reductions.

Language Model	H	PPL	WER
KN5	-	248.0	12.8
RNN+KN5	200	191.6	11.8
RNN-LDA+KN5	200	186.1	11.6
DRNN-LDA+KN5	200	**166.3**	**11.3**

Table 4. *Test PPL and WER results on 1 M words of WSJ corpus using class size 100.*

5. CONCLUSION AND FUTURE WORK

In this paper, we introduce a simple new LDA-based context RNNLM where the topic representation for a block of first L words is computed using individual document-based distribution over topics for each word in the block and then updated for each subsequent word using a traditional exponential decaying parameter. The new model is compared to a conventional LDA-based approximate topic representation approach that use corpus-based topic distribution of word. The proposed method yields best perplexity results on a well-known Penn Treebank corpus. This is because the approach incorporate proper context information as the document-based topic distribution of word could provide more topical information than the corpus-based topic distribution of word. The smaller value of the exponential parameter indicates that the proposed approach provides more significant context information than the traditional approach. Furthermore, we carried out 100-best rescoring experiment using WSJ corpus and reported WER improvements over the RNNLM and the RNN-LDA LM approaches. In the future, we will perform experiments using large amount of data.

6. REFERENCES

[1] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400-401, 1987.

[2] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, May 1995, pp. 181-184.

[3] R. Kuhn and R. D. Mori, "A cache-based natural language model for speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570-583, 1990.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, 1990.

[5] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of the Fifteenth Annual Conference on Uncertainty*

in Artificial Intelligence (UAI-99), July 30-August 1, 1999, pp. 289-296.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[7] J. R. Bellegarda, “Exploiting latent semantic information in statistical language modeling,” *IEEE Transactions on Speech and Audio Processing*, vol. 88, no. 8, pp. 1279-1296, 2000.

[8] D. Gildea and T. Hofmann, “Topic-based language models using EM,” in *Proc. of Sixth European Conference on Speech Communication and Technology (EUROSPEECH)*, September 1999, pp. 2167-2170.

[9] Y.-C. Tam and T. Schultz, “Unsupervised language model adaptation using latent semantic marginals,” in *Proc. of INTERSPEECH*, September 2006, pp. 2206-2209.

[10] T. Mikolov, M. Karafiat, L. Burget, J. H. Cernocky, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. of INTERSPEECH*, September 2010, pp. 1045-1048.

[11] 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.

[12] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* , vol. 6, no. 2, pp. 107-116, 1998.

[13] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp.157-166, 1994.

[14] T. Mikolov, “Statistical language models based on neural networks,” *PhD thesis, Brno University of Technology,* 2012.

[15] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu and M. A. Ranzato, “Learning longer memory in recurrent neural networks,” *arXiv preprint arXiv2:1412.7753*, 2015.

[16] M. Sundermeyer, R. Schlüter, and H. Ney, “LSTM neural networks for language modeling,” in *Proc. of INTERSPEECH*, September 2012, pp. 194-197.

[17] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” in *Proc. of IEEE Workshop on Spoken Language Technology (SLT)*, December 2012, pp. 234-239.

[18] X. Chen, T. Tan, X. Liu, P. Lanchantin, M. Wan, M. J. F. Gales, and P. C. Woodland, “Recurrent Neural Network Language Model Adaptation for Multi-Genre Broadcast Speech Recognition,” in *Proc. of INTERSPEECH*, 2015.

[19] T.-H. Wen, A. Heidel, H.-Y. Lee, Y. Tsao, and L.-S. Lee, “Recurrent neural network based personalized language modeling by social network crowdsourcing,” in *Proc. INTERSPEECH*, 2013.

[20] Y. Shi, “Language Models with Meta-information,” in *Ph.D. Thesis, Delft University of Technology*, 2014.

[21] O. Tilk and T. Alumae, “Multi-domain recurrent neural network language model for medical speech recognition,” in *Proc. HLT*, 2014.

[22] M. Steyvers and T. Griffiths, “Probabilistic topic models,” in *T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds),* Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007.

[23] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *Proc. of ICASSP*, May 2011, pp. 5528-5531.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” *Technical report, DTIC Document*, 1985.

[25] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of english: the penn treebank,” *Computational Linguistics*, vol. 19, no. 2, pp. 313-330, 1993.

[26] J. Garofolo, D. Graff, D. Paul, and D. Pallett, “CSR-I (WSJ0) complete LDC93S6A,” *Linguistic Data Consortium*, 1993.

[27] D. B. Paul and J. M. Baker, “The design for the wall street journal-based CSR corpus,” in *Proc. of International Conference on Spoken Language Processing (ICSLP)*, October 1992, pp. 899-902.

[28] T Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, “RNNLM-recurrent neural network language modeling toolkit,” in *Proc. of IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, December 2011, pp. 196-201.

[29] A. Stolcke, “SRILM-an extensible language modeling toolkit,” in *Proc. of ICSLP*, September 2002, pp. 901-904.

[30] S. Young, P. Woodland, G. Evermann, and M. Gales, “The HTK toolkit 3.4.1,” http://htk.eng.cam.ac.uk/, 2013.

[31] K. Vertanen, “HTK wall street journal training recipe,” http://www.inference.phy.cam.ac.uk/kv227/htk/