A Parallel Recurrent Neural Network for Language Modeling with POS Tags

Chao Su[1,3], **Heyan Huang**[1,2], **Shumin Shi**[1,2,*], **Yuhang Guo**[1], **Hao Wu**[1]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2]Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, Beijing 100081, China
[3]Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing 100048, China
{suchao,hhy63,bjssm,guoyuhang,wuhao123}@bit.edu.cn

Abstract

Language models have been used in many natural language processing applications. In recent years, the recurrent neural network based language models have defeated the conventional n-gram based techniques. However, it is difficult for neural network architectures to use linguistic annotations. We try to incorporate part-of-speech features in recurrent neural network language model, and use them to predict the next word. Specifically, we proposed a parallel structure which contains two recurrent neural networks, one for word sequence modeling and another for part-of-speech sequence modeling. The state of part-of-speech network helped improve the word sequence's prediction. Experiments show that the proposed method performs better than the traditional recurrent network on perplexity and is better at reranking machine translation outputs.[1]

1 Introduction

Language models (LMs) are crucial parts of many natural language processing applications, such as automatic speech recognition, statistical machine translation, and natural language generation. Language modeling aims to predict the next word given context or to give the probability of a word sequence in textual data. In the past decades, n-gram based modeling techniques were most commonly used in such NLP applications. However, the recurrent neural network based language model (RNNLM) and its extensions (Mikolov et al., 2010; Mikolov et al., 2011) have received a lot of attention and achieved the new state of the art results since 2010. The most important advantage of RNNLM is that it has the potential to model unlimited size of context, due to its recurrent property. That is to say, the hidden layer has a recurrent connection to itself at previous timestep.

Part-of-speech (POS) tags capture the syntactic role of each word, and has been proved to be useful for language modeling (Kneser and Ney, 1993; A. Heeman, 1998; Galescu and Ringger, 1999; Wang and Harper, 2002). Jelinek (1985) pointed out that we can replace the classes with POS tags in language model. Kneser and Ney (1993) incorporated POS tags into n-gram LM and got 37 percents improvement. But they got only 10 percents improvement with classes through clustering. A. Heeman (1998) redefined the objective of automatic speech recognition: to get both the word sequence and the POS sequence. His experiments showed 4.2 percent reduction on perplexity over classes.

It is common to build probabilistic graphical models using many different linguistic annotations (Finkel et al., 2006). However, the problem to combine neural architectures with conventional linguistic annotations seems hard. This is because neural architectures lack flexibility to incorporate achievements from other NLP tasks (Ji et al., 2016). To address the problem, (Ji et al., 2016) used a latent variable recurrent neural network (LVRNN) to construct language models with discourse relations. LVRNN was proposed by Chung et al. (2015) to model variables observed in sequential data.

*Corresponding author

[1]Our code is available at https://github.com/chao-su/prnnlm

31st Pacific Asia Conference on Language, Information and Computation (PACLIC 31), pages 140–147
Cebu City, Philippines, November 16-18, 2017

Inspired by the POS language models and the LVRNN models above, we use POS features to improve the performance of RNNLM. We assume that if we know the next POS tag, the search range to predict the next word will be shrinked; and the next POS is closely related with the POS sequence that has been seen before. Not the same as Ji et al. (2016), who used a latent variable to model the language annotation, we designed a parallel RNN structure, which consists two RNNs to model the word sequence and POS sequence respectively. And further the state of POS network has an impact on the word network.

In summary our main contributions are:

- We propose to model words and POS tags simultaneously by using a parallel RNN structure that consists of two recurrent neural networks, word RNN and POS RNN.
- We propose that the current state of the word network is conditioned on the current word, the previous hidden state, and also the state of POS network.
- We demonstrate the performance of our model by computing lower perplexity. We conducted our experiments on three different corpora, including Penn TreeBank, Switchboard, and BBC corpora.

The rest parts of this paper are organized as follows. Section 2 introduces the background techniques, including RNNLM and evaluation for language models. Section 3 elaborates our POS tag language model. Section 4 reports the experimental results. Section 5 reviews related work and Section 6 concludes the paper.

2 Background

In this section, we introduce the background techniques on which our work is based on. Recurrent neural network language models (RNNLMs) are important bases of our work. And the introduced evaluation method (perplexity) is used in this paper.

2.1 RNN Language Model

Mikolov et al. (2010) proposed to use recurrent neural network (RNN) to construct language model. By using RNN, context information can cycle inside the network for arbitrarily long time. Though it is also claimed that learning long-term dependencies by stochastic gradient descent can be quite difficult. We simply introduce Mikolov et al. (2010)'s recurrent neural network language model and its extensions (Mikolov et al., 2011) here.

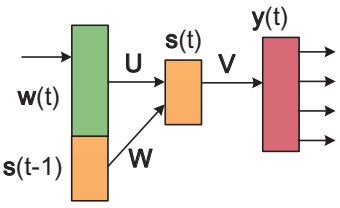

Figure 1: A simple Recurrent Neural Network.

We assume that a sentence consists of words, and each word is represented as $y(t)$, where t is current time step and $y(t) \in Vocab$. The architecture of RNNLM is shown in Fig. 1. Input to the network at time t is $\boldsymbol{w}(t)$ and $\boldsymbol{s}(t-1)$, where $\boldsymbol{w}(t)$ is a one hot vector representing the current word $y(t)$, and $\boldsymbol{s}(t-1)$ is the hidden layer $\boldsymbol{s}$ at previous time $t-1$. The hidden layer $\boldsymbol{s}(t)$ is the current state of the network. Output layer $\boldsymbol{y}(t)$ represents probability distribution of next word. Hidden and output layers are computed as:

$$s_i(t) = f\left(\sum_j w_j(t)u_{ij} + \sum_k s_k(t-1)w_{ik}\right) \quad (1)$$

$$y_k(t) = g\left(\sum_i s_i(t)v_{ki}\right) \quad (2)$$

where $f(z)$ is sigmoid activation function:

$$f(z) = \frac{1}{1+e^{-z}} \quad (3)$$

and $g(z)$ is softmax function:

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \quad (4)$$

In 2011, Mikolov et al. (2011) proposed some extensions of RNNLM. Those include a training algorithm for recurrent network called backpropagation through time (BPTT), and two speedup techniques. One is factorizing the output layer by class layer, and

the other is adding a compression layer between the hidden and output layers to reduce the size of the weight matrix $\boldsymbol{V}$. In this paper, we use two extensions, BPTT and class layer. But we still use the simple RNNLM architecture in figures for simplicity.

2.2 Evaluation

The quality of language models is evaluated both intrinsically by perplexity and extrinsically by quality of reranking machine translation outputs. The perplexity (PPL) of a word sequence $\boldsymbol{w}$ is defined as

$$PPL = \sqrt[K]{\prod_{i=1}^{K} \frac{1}{P(w_i|w_{1\ldots i-1})}} = 2^{-\frac{1}{K}\sum_{i=1}^{K}\log_2 P(w_i|w_{1\ldots i-1})} \quad (5)$$

Perplexity can be easily evaluated and the model which yields the lowest perplexity is in some sense the closest to the true model which generated the data.

Language model is an essential part of statistical machine translation systems, for measuring how likely it is that a translation hypothesis would be uttered by a native speaker (Koehn, 2010). Under the same conditions, a better language model brings a better translation system. Thus, we also evaluate our language model by evaluating the translation system who uses it. We use the most popular automatic evaluation metric for translation system, BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002); higher is better.

3 Parallel RNN LM with POS Feature

The traditional RNNLM models word sequences but ignores other linguistic knowledge. POS is such a kind of linguistic knowledge. It is easy to acquire with high annotation accuracy. We now present a parallel RNN structure over sequences of words and POS tag information. In this structure, we train two RNNs simultaneously, one for word sequence and another for POS sequence. We integrate the state of POS RNN with the word RNN.

3.1 Parallel RNN

The structure of the parallel RNN is shown in Fig. 2. The parallel RNN consists of two RNNs, word RNN and POS RNN. The word RNN is almost the same as the traditional RNN, except that its hidden state $\boldsymbol{s}(t)$ is also affected by an output from the state of POS RNN. The input layer of POS RNN consists two parts. One is the current POS tag $\boldsymbol{p}(t)$ and the other is the previous state of POS RNN. The hidden layer of POS RNN represents the current state of the network. The output layer represents the probability distribution of the next POS tag.

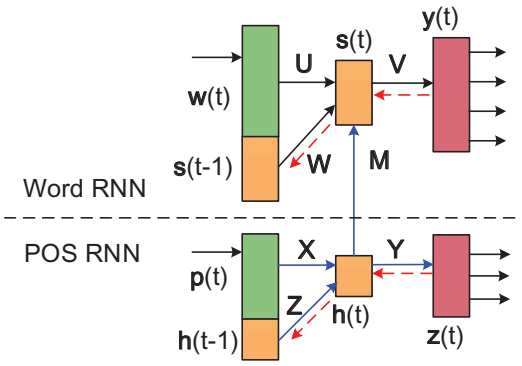

Figure 2: Structure of the Parallel RNN

We can see that the structure of the word RNN is similar with traditional RNN. The hidden layer of RNNLM theoretically contains all the information of the words those have been seen before. Similarly, the hidden layer of POS RNN contains the POS information in history. In order to use these information to predict the next word, we add a connection matrix between the hidden layers of word RNN and POS RNN.

In Fig. 2, the blue solid lines represent the forward computation, while the red dashed lines represent the back propagation of errors. Note that there is no error propagation from the hidden layer of word RNN to that of POS RNN. It is more likely that the latter affects the former like a latent variable in (Ji et al., 2016).

The hidden layer $\boldsymbol{h}(t)$ and output layer $\boldsymbol{z}(t)$ of POS RNN are computed as

$$h_i(t) = f\left(\sum_j p_j(t)x_{ij} + \sum_k h_k(t-1)z_{ik}\right) \quad (6)$$

$$z_k(t) = g\left(\sum_i h_i(t) y_{ki}\right) \tag{7}$$

The hidden layer of word RNN should be affected by that of POS RNN. So it is computed as

$$s_i(t) = f\left(\sum_j w_j(t) u_{ij} + \sum_k s_k(t) w_{ik} + \sum_l h_l(t) m_{il}\right) \tag{8}$$

3.2 Learning

In language model scenery, our purpose is to get the best word sequence. The training of the word RNN is the same as the traditional RNN. Though using the hidden layer of POS RNN to compute the state of the word RNN, we do not propagate the latter's error vector to the former. This is why we tend to treat the former also as a latent variable affecting the word sequence.

We train the POS RNN to maximize the log-likelihood function of the training data:

$$O = \sum_{i=1}^{T} \log d_{l_t}(t) \tag{9}$$

where T is the total number of POS tags in training examples, and l_t is the index of the correct POS tag for the t'th sample. The error vector in the output layer $\boldsymbol{e}_o(t)$ is computed as

$$\boldsymbol{e}_o(t) = \boldsymbol{d}(t) - \boldsymbol{z}(t) \tag{10}$$

where $\boldsymbol{d}(t)$ is the one-hot target vector that represents the POS tag at time t.

We update the parameters of POS RNN using stochastic gradient descent method. For example, the matrix Y is updated as

$$y_{jk}(t+1) = y_{jk}(t) + h_j(t) e_{ok}(t)\alpha - y_{jk}(t)\beta \tag{11}$$

where β is L2 regularization parameter. And the error vector propagated from the output layer to the hidden layer is

$$e_{hj}(t) = h_j(t)(1 - h_j(t)) \sum_i e_{oj}(t) y_{ij} \tag{12}$$

The update of the matrices X and Z is similar to equation (11). The error vector propagated from the hidden layer to its previous is similar to equation (12).

4 Experiments

We evaluated the proposed model in two ways: using perplexity (PPL) and reranking machine translation outputs.

4.1 Perplexity Setup

We evaluated our model on three corpora, including Switchboard-1 Telephone Speech Corpus (SWB), Penn TreeBank (PTB)[2], and BBC[3]. The former two corpora was used by Ji et al. (2016), while the last one was used by Wang and Cho (2016). We took all their work as comparisons. We splitted all the corpora into train, valid, and test sets, just like Ji et al. (2016) and Wang and Cho (2016) did. Statistics of the corpora are listed in Table 1. We tokenized all the corpora with tokenizer written by Pidong Wang, Josh Schroeder, and Philipp Koehn [4], and POS tagged with the Stanford POS Tagger [5].

We implemented our model based on Mikolov's RNNLM Tookit[6]. We considered the value 100 for the hidden dimension, and 10K for the vocabulary size.

The POS tagger's tagset consists of 48 tags. We counted the times of each tag appeared in the BBC corpus and sorted them in descending order (see Table 2). To verify the effect of POS tags, we gradually expanded our tagset's size (5, 10, 15, 20, 25, 30, 35, 40, 45) in the experiments. The size of POS RNN's hidden layer was set to one-fifth of the tagset's size. For example, $varsize = 40$ represents that we use the first 39 tags in Table 2 and reduce other tags to the $OTHER$ tag and the hidden size of POS RNN is set to be $40/5 = 8$.

[2] LDC97S62 for SWB, and LDC99T42 for PTB

[3] http://mlg.ucd.ie/datasets/bbc.html

[4] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[5] http://nlp.stanford.edu/software/tagger.shtml

[6] http://www.fit.vutbr.cz/~imikolov/rnnlm/

	SWB		PTB		BBC	
	#Sents	#Words	#Sents	#Words	#Sents	#Words
Train	211K	1.8M	37K	1M	37K	879K
Valid	3.5K	32K	3.6K	97K	2K	47K
Test	4.4K	38K	3.3K	91K	2.2K	51K

Table 1: Statistics of the Corpora SWB, PTB, and BBC

Order	POS	Times	Order	POS	Times
1	NN	121,359	21	“	11,010
2	IN	92,042	22	PRP$	8,939
3	NNP	88,331	23	”	7,961
4	DT	75,397	24	POS	7,711
5	JJ	52,851	25	:	5,219
6	NNS	47,003	26	FW	4,041
7	.	37,146	27	WDT	3,916
8	,	31,840	28	RP	3,583
9	VBD	31,575	29	JJR	2,990
10	VB	29,429	30	WP	2,865
11	RB	27,261	31	WRB	2,424
12	PRP	26,519	32	JJS	2,215
13	CC	22,554	33	NNPS	1,904
14	TO	22,440	34	EX	1,440
15	VBN	22,096	35	RBR	1,295
16	VBZ	20,795	36	$	1,127
17	CD	17,696	37	RBS	438
18	VBG	15,773	38	PDT	402
19	VBP	15,409	39	WP$	114
20	MD	11,015	OTHER		199

Table 2: Times of Each Tag Appeared in BBC Corpora

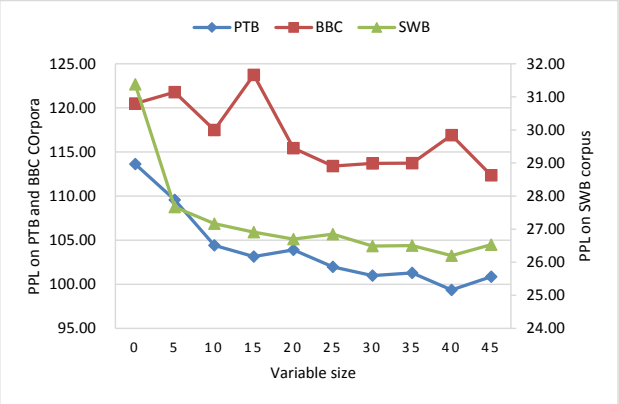

Figure 3: Perplexity Reduction with the Growth of Variable size

Model	SWB	PTB	BBC
5-gram	32.10	120.18	127.32
RNNLM	31.38	113.63	120.49
(Ji et al., 2016)	39.60	108.30	-
(Wang and Cho, 2016)	-	126.20	**105.60**
p-RNNLM	**26.20**	**99.36**	112.35
PPL reduction	16.5%	12.6%	6.8%

Table 3: Perplexity Comparison with Other Works

4.2 Perplexity Results

The perplexities of language modeling on the three corpora are summarized in Figure 3 and Table 3.

In Figure 3, we demonstrate the results using different number of most frequent POS tags, where the variable size is actually the size of POS RNN's hidden layer. Note that $varsize = 0$ represents a traditional RNNLM. We can see that the perplexity tends to reduce as the tagset size grows.

In Table 3, we compared our model with classic 5-gram model, Mikolov et al. (2010)'s RNNLM, Ji et al. (2016)'s, and Wang and Cho (2016)'s work. We can see that our parallel RNN (p-RNN) performs better than most of them except Wang and Cho (2016)'s work on BBC corpus. And our model gets 6.8%-16.5% PPL reduction over Mikolov et al. (2010)'s RNNLM.

4.3 MT Reranking Setup

We also performed reranking experiments on Chinese-English machine translation (MT) task. We evaluated the proposed parallel RNN language model by rescoring the 1000-best candidate translations produced by a phrase-based MT system. The decoder used was Moses(Koehn et al., 2007). The MT system was trained on FBIS (Foreign Broadcasting Information Service) corpus[7] containing about 250K sentence pairs and tuned with MERT (Minimum Error Rate Training) (Josef Och, 2003) on NIST MT02 test set. Our test sets included NIST

[7]LDC2003E14

MT 03, 04, and 05.

In reranking phase, we first performed MERT on two features, the MT score (got from MT system) and a LR score (the length ratio of the target language sentence to the source one), as a baseline. Both the RNNLM and p-RNNLM were trained on some news corpora[8] which contains about 2M sentences. We considered the values {100, 300, 500} for the hidden dimension of the word RNN, and 80K for the vocabulary size. We also performed POS tagging using the Stanford POS Tagger. We used the two trained models to rescore the 1000-best outputs from MT system and got RNNLM score and p-RNNLM score. Then we combine the two scores with MT score and LR score respectively to perform MERT to get their own weights. We tuned the weights for MT, LR, and RNNLM/p-RNNLM scores by using Z-MERT (Zaidan, 2009), which is a easy-to-use tool for MERT.

4.4 MT Reranking Results

The results for MT reranking is shown in Table 4. Both the RNN and p-RNN models outperform the baselines, Moses or MT+LR. The p-RNN model with 500 dimension size gets 0.59-1.04 BLEU improvement than MT+LR and at most 0.31 BLEU improvement than RNN model. Most of the improvements are statistically significant. The p-RNN model outperforms the RNN model on every test set with each dimension size.

5 Related Work

This paper draws on previous work language modeling including structured count-based and neural LMs.

5.1 Structured LMs

Efforts to incorporate linguistic annotations into language model include the structured LMs. Chelba et al. (1997) proposed a dependency language model using maximum entropy model. Chelba and Jelinek (1998) developed a language model that used syntactic structure to model long-distance dependencies. Charniak (2001) assigned the probability to a word conditioned on the lexical head of its parent constituent. Peng and Roth (2016) developed two models that captured semantic frames and discourse information.

POS-based LM originated from class-based LM (Jelinek, 1985; F. Brown et al., 1992), since POS tags captured the syntactic role of each word and could be seen as the equivalence classes. Kneser and Ney (1993) reported a perplexity reduction when combined their model with POS tags. A. Heeman (1998) redefined the speech recognition problem to find the best both word and POS sequences and incorporated POS-based LM.

5.2 Neural LMs

Bengio et al. (2003) proposed to use artificial neural network to learn the probability of word sequences. The feedforward network they used has to use fixed length context to predict the next word. Mikolov et al. (2010) used recurrent neural network to encode temporal information for contexts with arbitrary lengths.

In recent years, there was an increasing number of research integrating knowledge into RNN. Mikolov and Zweig (2012) incorporated topic information as a feature layer into RNNLM. Ji et al. (2015) employed the hidden states of the previous sentence as contextual information for predicting words in the current sentence. Ji et al. (2016) modeled discourse relation with Latent Variable Recurrent Neural Network (LVRNN) for language models. Ahn et al. (2016) proposed a language model which combined knowledge graphs with RNN. Dieng et al. (2016) proposed a TopicRNN to capture the global topic information for language modeling.

6 Conclusions

We proposed a parallel RNN structure to model both word and POS tag sequences. The structure consists of two RNNs, one for words and another for POS tags. The connection between the two network's hidden layers enabled the POS information to help to improve the word prediction. The role of POS RNN's hidden layer is similar to that of the latent variable in Ji et al. (2016)'s work. The perplexity of LM trained based on that structure got a reduction of

[8] LDC2003E14, LDC2000T46, LDC2007T09, LDC2005T10, LDC2008T06, LDC2009T15, LDC2010T03, LDC2009T02, LDC2009T06, LDC2013T11, LDC2013T16, LDC2007T23, LDC2008T08, LDC2008T18, LDC2014T04, LDC2014T11, LDC2005T06, LDC2007E101, LDC2002E18

System	MT02	MT03	MT04	MT05
Moses	28.09	24.38	28.03	24.19
MT+LR	28.07	24.40	28.11	24.26
MT+LR+RNN-100	28.25	25.16**	28.48**	24.39*
MT+LR+p-RNN-100	28.46**+	25.23**	28.70**++	24.53**+
MT+LR+RNN-300	28.57*	25.16**	28.72**	24.50**
MT+LR+p-RNN-300	28.62**+	25.26**	28.85**+	24.79**++
MT+LR+RNN-500	28.48**	25.38**	28.72**	24.59**
MT+LR+p-RNN-500	**28.66****	**25.44****	**28.84****	**24.90****

Table 4: MT Reranking Results. */**: significantly better than Moses ($p < 0.05/0.01$); +/++: significantly better than MT+LR+RNN ($p < 0.1/0.05$)

6.8%-16.5%. We used the LM to rerank MT outputs and got improvement on BLEU score.

Next, we will explore the expandability of the parallel RNN structure. We need to incorporate more linguistic knowledge to improve the performance of neural networks.

Acknowledgments

This work was supported by the National Basic Research Program (973) of China (No. 2013CB329303), the National Natural Science Foundation of China (Nos. 61132009, 61671064, 61502035), and Beijing Advanced Innovation Center for Imaging Technology (BAICIT-2016007).

References

Peter A. Heeman, 1998. *Sixth Workshop on Very Large Corpora*, chapter POS Tagging versus Classes in Language Modeling.

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *CoRR*, abs/1608.00318.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France.*, pages 116–123. Morgan Kaufmann Publishers.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada.*, pages 225–231. Morgan Kaufmann Publishers / ACL.

Ciprian Chelba, David Engle, Frederick Jelinek, Victor Jimenez, Sanjeev Khudanpur, Lidia Mangu, Harry Printz, Eric Ristad, Ronald Rosenfeld, Andreas Stolcke, and Dekai Wu. 1997. Structure and performance of a dependency language model. In *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25*. ISCA.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2980–2988.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John William Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *CoRR*, abs/1611.01702.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics, Volume 18, Number 4, December 1992*.

Rose Jenny Finkel, D. Christopher Manning, and Y. Andrew Ng, 2006. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, chapter Solving the Problem of Cascading Errors: Approximate Bayesian Inference for Linguistic Annotation Pipelines, pages 618–626. Association for Computational Linguistics.

Lucian Galescu and Eric K. Ringger. 1999. Augmenting words with linguistic information for n-

gram language models. In *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5-9, 1999*. ISCA.

F. Jelinek. 1985. Self-organized language modeling for speech recognition. *Technical Report*.

Yangfeng Ji, Trevor Cohn, Lingpeng Kong, Chris Dyer, and Jacob Eisenstein. 2015. Document context language models. *CoRR*, abs/1511.03962.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 332–342, San Diego, California, June. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Reinhard Kneser and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *Third European Conference on Speech Communication and Technology, EUROSPEECH 1993, Berlin, Germany, September 22-25, 1993*. ISCA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Philip Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5*, pages 234–239. IEEE.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048.

Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 5528–5531. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318. ACL.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 290–300, Berlin, Germany, August. Association for Computational Linguistics.

Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1329. Association for Computational Linguistics.

Wen Wang and Mary P. Harper. 2002. The superarv language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 238–247, Stroudsburg, PA, USA. Association for Computational Linguistics.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.