Hierarchical Recurrent Neural Network for Document Modeling

Rui Lin[†]*, **Shujie Liu**[‡], **Muyun Yang**[†], **Mu Li**[‡], **Ming Zhou**[‡], **Sheng Li**[†]
[†]Harbin Institute of Technology
{linrui,ymy}@mtlab.hit.edu.cn lisheng@hit.edu.cn
[‡]Microsoft Research
{shujliu,muli,mingzhou}@microsoft.com

Abstract

This paper proposes a novel hierarchical recurrent neural network language model (HRNNLM) for document modeling. After establishing a RNN to capture the coherence between sentences in a document, HRNNLM integrates it as the sentence history information into the word level RNN to predict the word sequence with cross-sentence contextual information. A two-step training approach is designed, in which sentence-level and word-level language models are approximated for the convergence in a pipeline style. Examined by the standard sentence reordering scenario, HRNNLM is proved for its better accuracy in modeling the sentence coherence. And at the word level, experimental results also indicate a significant lower model perplexity, followed by a practical better translation result when applied to a Chinese-English document translation reranking task.

1 Introduction

Deep Neural Network (DNN), a neural network with multiple layers, has been proven powerful in many different domains, such as visual recognition (Kavukcuoglu et al., 2010) and speech recognition (Dahl et al., 2012), ever since Hinton et al. (2006) formulated an efficient training method for it.

In addition to the applications mentioned above, many neural network based methods have also been applied to natural language processing (NLP) tasks with great success. For example, Collobert et al. (2011) propose a generalized DNN framework for a variety of fundamental NLP tasks, including part-of-speech tagging (postag), chunking, named entity recognition (NER), and semantic role labeling.

DNN is successfully introduced to do word-level language modeling, aka., to predict the next word given the history words. Bengio et al. (2003) propose a feedforward neural network to train a word-level language model with a limited n-gram history. To leverage as much history as possible, Mikolov et al. (2010) apply recurrent neural network to word-level language modeling. The model absorbs one word each time, keeps the information in a history vector, and predicts the next word with all the word history in the vector.

Word-level language model can only learn the relationship between words in one sentence. For sentences in one document which talks about one or several specific topics, the words in the next sentence are chosen partially in accordance with the previous sentences. To model this kind of coherence of sentences, Le and Mikolov (2014) extend word embedding learning network (Mikolov et al., 2013) to learn the paragraph embedding as a fixed-length vector representation for paragraph or sentence. Li and Hovy (2014) propose a neural network coherence model which employs distributed sentence representation and then predict the probability of whether a sequence of sentences is coherent or not.

In contrast to the methods mentioned above which learn the word relationship in or between the sentences separately, we propose a hierarchical recurrent neural network language model (HRNNLM) to capture the word sequence across the sentence boundaries at the document level. HRNNLM is essentially a combination of a word-level language model and a sentence-level language model, both of which are recurrent neural networks. The word-level recurrent neural network follows (Mikolov et al., 2010). The sentence-level language model is another recurrent neural network that takes sentence represen-

*Contribution during internship at Microsoft Research.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 899–907,
Lisbon, Portugal, 17-21 September 2015.

tation as input, and predicts the words in the next sentence. Similar to (Mikolov et al., 2010), the hidden layer in the sentence-level recurrent neural network contains the sentence history information. The hidden layer containing the history information of previous sentences is then linked as an input to the word-level recurrent neural network to predict the next word together with the word-level history vector. This allows the language model to predict the next word probability distribution beyond the words in the current sentence.

We propose a two-step training approach to optimize the parameters of HRNNLM. In the first step, we train the sentence-level language models independently . And then, we connect the hidden layer of the sentence-level language model to the input of word-level RNNLM and train the two models jointly until converged. At sentence level, we evaluate our model with a sentence ordering task and the result shows our method can outperform a maximum entropy based and another state-of-the-art solution. At word level, we compare our method with the conventional recurrent neural network based language model, finding the perplexity is reduced significantly. We also apply our method to rank machine translation output and conduct experiments on a Chinese-English document translation task, yielding a better translation results compared with a state-of-the-art baseline system.

The rest of this paper is organized as follows: Section 2 introduces work related to applying neural network to document modeling and SMT. Section 3 introduces the general framework for document modeling. Our sentence-level language model and its training is described in Section 4, and the overall HRNNLM and its training is presented in Section 5. Section 6 presents our experiments and their results. Finally, we conclude in Section 7.

2 Related work

In this section, we introduce previous efforts on applying neural network to model words coherence across sentence boundaries as well as works on improving machine translation performance at discourse level.

Mikolov and Zweig (2012) propose a RNN-LDA model to implement a context dependent language model. They augment the contextual information into the conventional RNNLM via a real-valued input vector, which is the probability distribution computed by LDA topics for using a block of preceding text. They train a Latent Dirichlet Allocation (LDA) model using documents consisting of about 10 sentences long text from Penn Treebank (PTB) training data. Their approach outperforms RNNLM in perplexity on PTB data with a limited context history over topics instead of complete information of preceding sentences.

Le and Mikolov (2014) extend the Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model (Skip-gram) (Mikolov et al., 2013) by introducing a paragraph vector. In their method, the paragraph vector is learnt in a similar way of word vector model, and there will be $N \times P$ parameters, if there are N paragraphs and each paragraph is mapped to P dimensions. Different from them, the sentence vectors of our model are learnt with nearly unlimited sentence history based on a RNN framework, in which, bag of words in the sentence are used as input. The sentence vector is no longer related with the sentence id, but only based on the words in the sentence. And our sentence vector also integrates nearly all the history information of previous sentences, while their model cannot.

Li and Hovy (2014) implement a neural network model to predict discourse coherence quality in essays. In their work, recurrent (Sutskever et al., 2011) and recursive (Socher et al., 2013) neural networks are both examined to learn distributed sentence representation given pre-trained word embedding. The distributed sentence representation is assigned to capture both syntactic and semantic information. With a slide window of the distributed sentence representation, a neural network classifier is trained to evaluate the coherence of the text. Successful as it is in scoring the coherence for a given sequence of sentences, this method is attempted to discriminate the different word order within a sentence.

An attempt of introducing RNN into convolutional neural network (CNN) is investigated by (Xu and Sarikaya, 2014) for spoken language understanding (SLU). To alleviate more contextual information, they apply a CNN with Jordan-type (Jordan, 1997) recurrent connections. The recurrent connections send the distribution of the last softmax layer's output to the current input layer as additional features. Aimed to improve SLU domain classification, their model is essentially a kind of document representation with certain text

information, neglecting the coherence information between sentences.

Following the thread modeling the word sequence relationship within and across sentences, we propose a hierarchical recurrent neural network language model consist of a sentence-level language model and a word-level language model. This overall network is trained to capture the coherence between sentences and predict words sequence with preceding sentence contexts.

For statistical machine translation (SMT) in which we checked out model as a scenario, DNN has also been revealed for certain good results in several components. Yang et al. (2013) adapt and extend the CD-DNN-HMM (Dahl et al., 2012) model to the HMM-based word alignment model. In their method, they use bilingual word embedding to capture the lexical translation information and modeling the context with surrounding words. Liu et al. (2014) propose a recursive recurrent neural network (R^2NN) for end-to-end decoding to help improve translation quality. And Cho et al. (2014) propose a RNN Encoder-Decoder which is a joint recurrent neural network model at the sentence level as conventional SMT decoder does. However, at the discourse level, there is little report on applying DNN to boost the translation result of a document.

3 Document Language Modeling

Statistical language model assigns a probability to a natural language sequence. Conventional language models only focus on the word sequence within a sentence. For sentences in one document talking about one or several specific topics, the adjacent sentences should be in a coherent order. Therefore, the words in the next sentence are also dependent on the preceding sentences. To model the coherence of sentences in the document D, which contains N sentences S_1, S_2, S_3, ..., S_N, we need to maximize the objective as follow:

$$\begin{aligned} p(D) =& p(S_1, S_2, ..., S_N) \\ =& p(S_1) \cdot p(S_2|S_1) \cdot p(S_3|S_1, S_2) \\ & ...p(S_N|S_1, S_2, ..., S_{N-1}) \end{aligned} \quad (1)$$

For the sentence S_k containing words w_1, w_2, w_3, ..., w_T, $p(S_k|S_1, S_2, ..., S_{k-1})$ is defined as:

$$\begin{aligned} & p(S_k|S_1, S_2, ..., S_{k-1}) \\ & = p(w_1, w_2, ..., w_T|S_1, ..., S_{k-1}) \\ & = p(w_1|S_1, ..., S_{k-1}) \cdot p(w_2|w_1, S_1, ..., S_{k-1}) \\ & ...p(w_T|w_1, w_2, ..., w_{T-1}, S_1, ..., S_{k-1}) \end{aligned} \quad (2)$$

As a special case of approximation to this, classical n-gram language model keep only several words as history, discarding any information across the sentence boundaries. Recurrent neural network language model (Mikolov et al., 2010) uses a hidden layer which employs a real-valued vector recurrently as network's input to keep as many history as possible. This makes RNNLM be able to extend for capturing history beyond a sentence.

To prevent the potential exponential decay of the history, the history length in RNN can not be too long. Here we approximate the history information of previous sentences, $p(S_k|S_1, S_2, ..., S_{k-1})$, by the following:

$$\begin{aligned} & p(S_k|S_1, S_2, ..., S_{k-1}) = \\ & p(BoW_{S_k}|BoW_{S_1}, ..., BoW_{S_{k-1}}) \cdot p(S_k|BoW_{S_k}) \end{aligned} \quad (3)$$

where BoW_{S_k} denotes the bag of words for the sentence S_k. The document is thus generated in two steps.

- Given the previous sentences BoW_{S_1} ,..., $BoW_{S_{k-1}}$ (treating them as bag of words here), first generate the words which will show in the next sentence without considering their order with $p(BoW_{S_k}|BoW_{S_1}, ..., BoW_{S_{k-1}})$
- Generate the words one by one with $p(S_k|BoW_{S_k})$.

The first phase actually completes sentence-level language modeling, and the second addresses the word-level language modeling. Because recurrent neural network has a natural advantage in processing sequential data, we investigate how to model the whole process under a unified framework of recurrent neural network.

4 Sentence-level Language Model

In this section, we describe how to leverage recurrent neural network for sentence-level language modeling. Mikolov et al. (2010) demonstrate a recurrent neural network language model (RNNLM)

for word ordering. It overcomes the limitations of classical language model in capturing only a fixed-length history, yielding a significant performance improvements in terms of perplexity reduction and speech recognition accuracy. Here we adept this framework for a RNN based sentence-level language modeling, i.e. RNNSLM.

4.1 Model

A conventional language model reads a word each time, keeps several words as history and then predict the probability distribution of the next word. Similar to this, our sentence-level language model reads a sentence which is a bag of words representation. And then it stores the sentence history which captures coherence of sentences in a real-valued history vector. With the history vector, our model can predict which words are most likely to appear in the next sentence. All these will be modeled by a recurrent neural network.

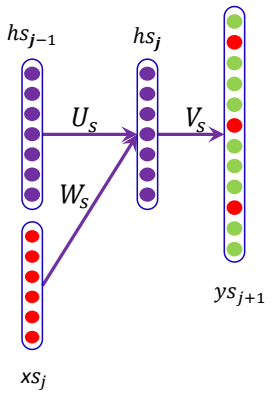

Figure 1: Recurrent Neural Network for Sentence-level Language Modeling

As shown in Figure 1, similar to the conventional recurrent neural network, for the sentence j, our network has two input layers xs_j and hs_{j-1}. xs_j is the current sentence representation, and hs_{j-1} is the history information vector before sentence j. The model has a hidden layer hs_j, which will combine the history information of hs_{j-1} and the current sentence input xs_j, and an output layer ys_{j+1}, which generates the probabilities of the words in the sentence $j+1$. The layers are computed as follows:

$$hs_j = f(U_s \cdot hs_{j-1} + W_s \cdot xs_j) \quad (4)$$

$$ys_{j+1} = g(V_s \cdot hs_j) \quad (5)$$

where W_s, U_s and V_s denote the weight matrix. $f(z)$ is a $HTanh$ function:

$$f(z_j) = \begin{cases} -1 & z_j < -1 \\ z & -1 < z_j < 1 \\ 1 & z_j > 1 \end{cases} \quad (6)$$

and $g(z)$ is a softmax function:

$$g(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (7)$$

The output layer ys_{j+1} is a $1 \times V$ vector that represents probability distribution of words in the next sentence given the current sentence xs_j and previous history hs_{j-1}, where V denotes vocabulary size.

To emphasize coherence between the adjacent sentences, we further add some bigram-like bag of words feature to the output layer. As mentioned in (Mikolov, 2012), this is kind of maximum entropy feature which can be derived by a two-layer neural network. Some experiments show that perplexity significantly decreases after adding these features. Following (Mikolov, 2012), where, the maximum entropy bigram features are added to our RNNSLM by a direct connection between the feature input array and output layer ys_{j+1}. Following (Mahoney, 2000), we map bigram maximum entropy features to a fixed-length array to reduce the memory complexity of direct connections with feature hashing. Then the output layer can be computed as follow:

$$ys_{j+1}(t) = g(V_s(t) \cdot hs_j + \sum_{w \in xs_j} D_{hash(w,t)}) \quad (8)$$

where (t) denotes the t-th row of a vector or a matrix. D denotes that the hash array contains feature weights and $hash(w_i, w_j)$ denotes the hash function for mapping bigram features to a fixed-length array. For a output ys_{j+1}, multiple connections may be activated according to the words in sentence xs_j.

4.2 Training

The training objective of our RNNSLM is to find the best parameters for predicting the words of next sentence. Formally, given the next sentence S_k containing words w_1, w_2, w_3, ..., w_T. The training objective according to (Mikolov et al.,

2013) can be denoted by:

$$log(p(BoW_{S_k}|BoW_{S_1}, ..., BoW_{S_{k-1}})) = \frac{1}{T}\sum_{t=1}^{T} logp(w_t|BoW_{S_1}, ..., BoW_{S_{k-1}}) \quad (9)$$

For weight matrix W_s, U_s, V_s and hash feature weight D, the parameter are trained similar to the conventional recurrent neural network. The learning rate α is set to 0.1 at the start of the training as suggested in (Mikolov et al., 2010). After each epoch, it can be determined by the training loss of network. If the loss decreases significantly, training continues with the same learning rate. Otherwise, if the loss increases, the training will be executed with a new learning rate $\alpha/2$. The training process will be terminated after about 30 epochs.

4.3 Initialization

All elements in weight matrix W_s and U_s are initialized by randomly sampling from a uniform distribution $[-\frac{1}{K_1}, \frac{1}{K_1}]$, where K_1 is the size of the input layer. Elements in weight matrix V_s are initialized by randomly sampling from a uniform distribution $[-\frac{1}{K_2}, \frac{1}{K_2}]$, where K_2 denotes the size of the hidden layer. The hash feature weight array D is initialized as 0.

For the initialization of hs_0, it can be set to a vector of the same values, which is 0.1.

5 Hierarchical Recurrent Neural Network

In the previous section, we propose a RNNSLM which models the coherence between sentences but ignores the word sequence within a sentence. Ideally, a perfect document model should not only capture the information between sentences but also the information with sentence. So we propose a hierarchical recurrent neural network language model (HRNNLM) to fulfill this issue.

5.1 Model

A hierarchical recurrent neural network consists of two independent recurrent neural network. For a conventional word-level language model, it predict the next word only using the word history within the sentence. To capture the longer history, we integrate the sentence history into the word-level language model from sentence-level language model, which forms a hierarchical recurrent neural network.

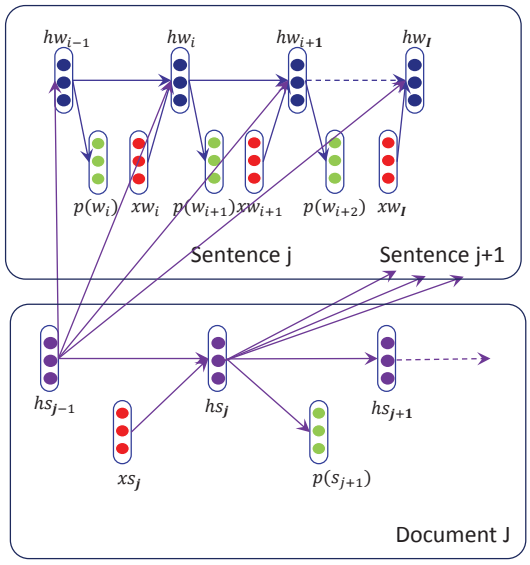

Figure 2: Hierarchical recurrent neural network

As illustrated Figure 2, the upper part is the unfolded illustration of conventional recurrent neural network based language model. It takes one word w_i each time with the previous history information hw_{i-1} together and predicts the probability of the next word $p(w_{i+1})$ with the information kept in the history vector hw_i. The lower part is our RNNSLM, which takes the bag of words representation of a sentence xs_j each time with the history information of previous sentences hs_{j-1} together and predicts the bag of words in the next sentence $p(s_{j+1})$ with the information kept in hs_j.

We integrate these two recurrent neural networks together by adding connections between the sentence-level history vector hs_{j-1} and word level history vector hw_i. So while predicting the next word w_{i+1} of the current sentence, our model will consider the current word w_i, history of previous sentences hs_{j-1} and history of previous words hw_{i-1}. The new word-level history vector hw_i is computed as:

$$hw_i = f(U_w \cdot hw_{i-1} + W_w \cdot xw_i + U_{sw} \cdot hs_{j-1}) \quad (10)$$

where $f(z)$ is a $HTanh$ function. For HRNNLM, we also add a bigram hash feature, similar as we do for RNNSLM.

5.2 Training

The HRNNLM can be trained from scratch following Mikolov et al. (2010) with a dual objective. But this is not without problem. Beginning

of training phase, the sentence history is unstable since the parameters of sentence-level language model are kept updating. Consequently, the training of HRNNLM will be also unstable and hard to converge with unstable sentence history.

In this paper, we approximate the whole training of HRNNLM by a two-step training method. We first train a RNNSLM until it converges. Then we connect the hidden layer of RNNSLM to the hidden layer of RNNWLM. To increase the training speed, all the parameters of RNNSLM are fixed while training HRNNLM. We only update the random initialized parameters in HRNNLM, though ideally the gradient of the sentence history vector could change and the RNNSLM could be updated again. The learning rate α is set to 0.1 and the updating of learning rate is the same as suggested in Section 4.2. All the parameters can be initialize as suggested in Section 4.3.

6 Experiments

We evaluate the sentence-level performance of HRNNLM by the common coherence evaluation of sentence ordering task, its word-level performance by perplexity measure. We also apply our HRNNLM to SMT reranking task in an open Chinese-English translation dataset. The translation performance index is the IBM version of BLEU-4 (Papineni et al., 2002).

6.1 Sentence Ordering

We follow (Barzilay and Lapata, 2008) to evaluate our sentence-level language model via a sentence ordering task with test set 2010 (tst2010), 2011 (tst2011) and 2012 (tst2012) from IWSLT 2014, totaling 37 English documents. 20 random permutations of sentences for each document are generated. Each permutation and its original document are combined as an article pair. Our goal is to find the original one among all the article pairs.

The training data for sentence-level language model is the 1,414 English documents from the parallel corpus also provided by the IWSLT 2014 spoken language translation task. 90% of the documents are for training and the rest are reserved for validation. The size of the hidden layer is set to 30 and hash array size is 10^7.

We define the log probability of a given document as its coherence score. The document with the higher score is regarded as the original document.

We provide two baselines for sentence ordering. One is the state-of-the-art recursive neural network based method proposed by (Li and Hovy, 2014). We implement their model trained and tested with our data. The other is a maximum entropy classifier trained with bag of words features of adjacent sentences which can generate a coherent probability of adjacent sentences. The document with the higher sum of log probability for each adjacency sentences is regarded as the original document. Table 1 shows the accuracy of our system and baseline.

Setting	Accuracy
Recursive	91.39%
ME system	91.89%
Our system	95.68%

Table 1: Accuracy of the sentence ordering task for each system

From Table 1 we can see that the maximum entropy model and the recursive neural network model has almost the same performance. Compared with the baseline systems, the proposed HRNNLM achieves significant improvement with nearly 4.3% improvement in term of accuracy. The experimental result shows that the HRNNLM can model document coherence and capture cross-sentence information.

6.2 Word-level Model Perplexity

We compare the word level performance of HRNNLM with the most popular RNNLM in terms of model perplexity. For a fair comparison, we follow (Mikolov et al., 2010) and train the model also on 90% of the 1.414 English documents form IWSLT 2014, totaling about 3M words. Then we train our model with the same hidden layer size and hash array size as the baseline system. The perplexity of these two models is evaluated on held-out documents, about 370K words. The results are shown in Table 2.

Setting	Perplexity
RNNLM-30	183
HRNNLM-30	174

Table 2: Perplexity of the different language model

According to Table 2, it is reasonable to claim that, by integrating history information of previous

sentences, the model perplexity decreased significantly. Empirically, this confirms the hypothesis that the words selection for the next sentence is dependent on its preceding sentences in the same document.

6.3 Spoken Language Translation

The conventional SMT systems translate sentences independently, without considering the coherence of the sentences in the same document. In order to learn translation coherence between sentences, we apply the HRNNLM to machine translation reranking task.

6.3.1 Data Setting and Baselines

The data comes from the IWSLT 2014 spoken language translation task. The training data consists of 1,414 documents on TED talks, and contains 179k sentence pairs, about 3M Chinese words, and 3.3M English words. The language model for SMT is a 4-gram language model trained with the English documents in the training data. The development set is specified by IWSLT as dev2010, and the test set contains 37 documents from tst2010, tst2011 and tst2012.

The IWSLT 2014 baseline system is built upon the open-source machine translation toolkit Moses at the default configuration, proposed by (Cettolo et al., 2012). We also train a decoder, which is an in-house Bracketing Transduction Grammar (BTG) (Wu, 1997) in a CKY-style decoder with a lexical reordering model trained with maximum entropy (Xiong et al., 2006). The decoder uses commonly used features, such as translation probabilities, lexical weights, a language model, word penalty, and distortion probabilities.

6.3.2 Rerank System

Our reranking system is a linear model with several features, including the SMT system final scores, sentence-level language model scores, and HRNNLM scores. It should be noted all these features are actually employed by the SMT model except for the HRNNLM score. Since Minimum Error Rate Training (MERT) (Och, 2003) is the most general method adopted in SMT systems for tuning, the feature weights are fixed by MERT.

For our reranking system, to score the translation of one sentence we need the translation results of all the previous sentences in the document. Our SMT decoder generates 10-best results of all the sentences of the documents and the reranking system select the best translation result for the first sentence at first. With the translation of first sentence, we score all the translation candidates of the second sentence and select the best one as the result. Following this procedure, we can get the translation results for all the sentences in the document.

6.3.3 Results

The HRNNLM focus on exploiting longer context, esp. cross-sentence word dependencies. Therefor the translation data for IWSLT 2014 is organized as documents instead of sentences for our rerank system. We hope HRNNLM will enable a context-sensitive reranking process, capturing the syntactic and logic relationships between the sentences in the same document.

Setting	tst2010	tst2011	tst2012
IWSLT	11.12	13.34	-
Baseline	12.40	15.09	13.52
SMT + Rerank	12.55	15.23	13.70

Table 3: BLEU scores of SMT systems. The IWSLT is a public baseline which issued by the organizer of IWSLT 2014, as described in (Cettolo et al., 2012).

The translation performance comparison is shown in Table 3. From Table 3, we can find that the rerank system improves SMT performance consistently. For a single sentence without the context information, there are several appropriate translations and it is hard to tell which one is better. When considering the context of a document (previous sentences for our model), some translation candidates may not be coherent with the others which should not be selected. Our model can generate the most coherent translation results by considering previous sentence history.

For example, we have the following two Chinese sentence in one document together with their correct translation:

我 拍摄 过 的 冰山, 有些 冰 是 非常 年轻 - - 几千 年 年龄
Some of the ice in the icebergs that I photograph is very young - - a couple thousand years old.
有些 冰 超过 十万 年
And some of the ice is over 100,000 years old.

Chinese word " 有些" means "some" in English. But when it is used in parallelism sentences, it means "some of" instead of "some". The traditional SMT system translates the italics part without considering the context. The translation result for this kind of system is:

Some ice more than 100,000 years.

For our system, the HRNNLM can take previous sentence as context and learn the parallelism between the two sentences. It can select the best translation "some of" for 有些, and the output of our system is:

Some of the ice more than 100,000 years.

We also calculate the BLEU increase ratio of our system on document level. The ratio is defined as $\frac{1}{N}\#(BleuD_{rerank} > BleuD_{baseline})$, where N denotes the number of documents, and $\#(BleuD_{rerank} > BleuD_{baseline})$ denotes the number of documents for which document level BLEU score of reranking system is higher than the baselines. The results are shown in Table 4.

tst2010	tst2011	tst2012
72.73%	71.43%	75%

Table 4: Experimental results to test BLEU increase ratio after reranking

From Table 4, we can find that, for all the three test data sets, our reranking system can achieve better performance for more than 70% documents.

7 Conclusion and Future Work

In this paper, we propose a hierarchical recurrent neural network language model for document modeling. We first built a RNNSLM to capture the information between sentences. Then we integrate the hidden layer of RNNSLM into the input layer of word-level language model to form a hierarchical recurrent neural network. This enables the model be able to capture both in-sentence and cross-sentence information in a unified RNN. Compared with conventional language models, our model can perceive a longer history than other language models and captures the context patterns in the previous sentences. At sentence level, we examine our model with sentence ordering task. At word level, we test the model perplexity. We also conduct a SMT rerank experiment on IWSLT 2014 data set. All these experimental results show that our hierarchical recurrent neural network has a satisfying performance.

In the future, we will explore better sentence representation such as distributed sentence representation as input for our sentence-level language model to better model document coherence. We can even update the gradient from different RNN to get a better performance.

Acknowledgments

We are grateful to the three anonymous reviewers for their helpful comments and suggestions. We also thank Dongdong Zhang, Lei Cui for useful discussions. This paper is supported by the project of National Natural Science Foundation of China (Grant No. 61272384, 61370170 &61402134).

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

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

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. pages 1724–1734, October.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42.

Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. *Advances in psychology*, 121:471–495.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. 2010. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. pages 1188–1196.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048. Association for Computational Linguistics.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation. In *Proceedings of ACL*, pages 1491–1500.

Matthew V Mahoney. 2000. Fast text compression with neural networks. In *FLAIRS Conference*, pages 230–234.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Spoken Language Technology Workshop, IEEE*, pages 234 – 239.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomáš Mikolov. 2012. *Statistical language models based on neural networks*. Ph.D. thesis, Ph. D. thesis, Brno University of Technology.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 521–528. Association for Computational Linguistics.

Puyang Xu and Ruhi Sarikaya. 2014. Contextual domain classification in spoken language understanding systems using recurrent neural network. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 136–140. IEEE.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 166–175, Sofia, Bulgaria, August. Association for Computational Linguistics.