Deep Word Association: A Flexible Chinese Word Association Method with Iterative Attention Mechanism

Yaoxiong Huang[1], Zecheng Xie[1], Manfei Liu[1], Shuaitao Zhang[1], and Lianwen Jin[1,2](✉)

[1] School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China
hwang.yaoxiong@gamil.com, zcheng.xie@gamil.com, manfei.l.liu@gamil.com,
z.shuaitao@gamil.com, lianwen.jin@gamil.com

[2] SCUT-Zhuhai Institute of Modern Industrial Innovation,
South China University of Technology, Zhuhai, China

Abstract. Word association is to predict the subsequent words and phrase, acting as a reminder to accelerate the text-editing process. Existing word association models can only predict the next word inflexibly through a given word vocabulary or a simply back-off N-gram language model. Herein, we propose a deep word association system based on attention mechanism with the following contributions: (1) To the best of our knowledge, this is the first investigation of an attention-based recurrent neural network for word association. In the experiments, we provide a comprehensive study on the attention processes for the word association problem; (2) An novel approach, named DropContext, is proposed to solve the over-fitting problem during attention training procedure; (3) Compared with conventional vocabulary-based methods, our word association system can generate an arbitrary-length string of words that are reasonable; (4) Given information on different hierarchies, the proposed system can flexibly generate associated words accordingly.

Keywords: Word association · Attention mechanism
Recurrent neural network · Chinese · DropContext

1 Introduction

Given a word, phrase, or sentence of arbitrary length, word association requires a machine to predict the following word, phrase, or even sentence that the user would like to express, acting as a reminder to accelerate the text-editing process. Word association is widely used in daily life, such as text input to smartphones, the auto-fill of fields in a web browser, and question/answer systems, which can not only save time and effort but also prevent spelling errors by providing users with a list of the most relevant words. Specifically, when a word is input by a user, the word association system provides a list of candidate words for the user

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 112–123, 2018.
https://doi.org/10.1007/978-3-030-03338-5_10

to select and then updates the associated word list until the user has finished the text editing task.

In the community, methods have been presented for the advancement of word association. Generally, custom systems use a vocabulary or statistical information for word association. PAL [1], the first word association system, predicted the most frequent words that match the given words, completely ignoring any useful context information. Profet [2] (for Swedish) and WordQ [3] (for English) used both word unigrams and bigrams to improve the word association but still suffered from a lack of context information, which would easily lead to syntactically inappropriate words. Considering the inflexibility of the above-mentioned systems, an approach that models the complex context information of the given words is significantly important for the word association problem. In recent years, neural networks [4–6] have demonstrated outstanding ability in language models (LMs). In particular, recurrent neural network LMs (RNNLMs) [7] use long-term temporal dependencies without a strong conditional independence assumption. As RNNLMs become more popular, Sutskever et al. [8] developed a simple variant of the RNN that can generate meaningful sentences by learning from a character-level corpus. Zhang and Lapata [9] have conducted some interesting work and use RNNs to generate Chinese poetry. Furthermore, the ability to train deep neural networks provides a more sophisticated method of exploiting the underlying context information of the sentence, thereby making the prediction more accurate [10].

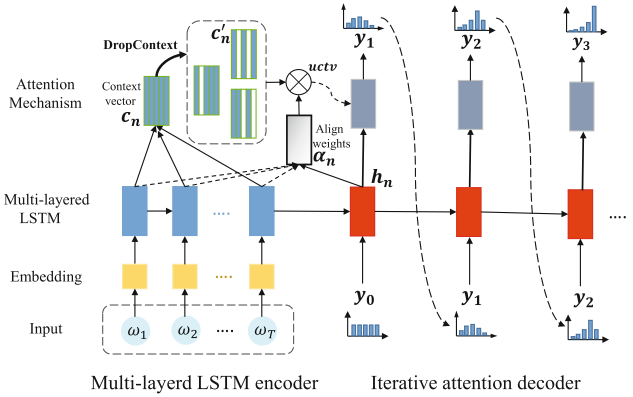


Fig. 1. The proposed word association system consists of two parts: (1) a multi-layered LSTM encoder that learns a hierarchy of semantic features from the input text corpus $\boldsymbol{w} = w_1, \cdots, w_T$. and (2) an iterative attention decoder module (with DropContext) that iteratively updates attentions and refines current predictions. Note that y_0 is uniform distribution and y_N predicts the finally results.

LSTM has the ability to remember the past information, but it is quite limited and thus easily leads to prediction failure [11]. Therefore, the attention mechanism has gained popularity recently in training neural networks [12]; it allows models to learn the alignments between different modalities. The alignments may be between the frame level and text in the speech recognition task [13], or between the source words and translation in the neural machine translation problem [14], allowing the network focus more on the important part of the input. To the best of our knowledge, it is the best choice for natural language processing, e.g., word association problem.

The performance of the current neural network is highly dependent on the greedy learning of model parameters via many iterations based on a properly designed network architecture [15]. During the training phase, it is easy to encounter a problem of over-fitting. Many previous works have been dedicated to solving this problem, e.g., Dropout [16] and DropConnect [17]. Nevertheless, they were not appropriate for the attention mechanism.

Inspired by the aforementioned papers and works, we proposed a word association system that integrates multi-layered LSTM with iterative attention mechanism. The primary contributions of the network can be summarized as follows:

- Attention mechanism is integrated to allow the proposed system to iteratively review context information as well as historical prediction.
- A novel training strategy, namely DropContext, is proposed to alleviate the over-fitting problem during the learning process.
- Given certain information of different hierarchies, the network can generate words of arbitrary length, flexibly. The richer the information provided, the more meaningful words are associated.
- The effectiveness of the proposed system is validated not only by word association on huge Chinese corpus, but also by a poem generating experiment.

The remainder of this paper is organized as follows: Sect. 2 presents a system overview. Section 3 describes the results and performance evaluation of our proposed model. Section 4 summarizes our work.

2 System Overview

Given the training text corpus $\boldsymbol{w} = w_1, \cdots, w_T$ in V, where V is the word dictionary, our word association system f, aims to minimize the loss function $L(\boldsymbol{w})$ as the negative log probability of correctly predicting all the associated words in the text corpus:

$$L(\boldsymbol{w}) = -\frac{1}{T}\sum_{t} logf(w_t, w_{t-1}, \cdots, w_{t-n+1}; \theta) + R(\theta) \tag{1}$$

where T is the total length of the corpus and $R(\theta)$ is a regularization term. Figure 1 describe the detailed architecture of our word association system. Given the training corpus $\boldsymbol{w} = w_1, \cdots, w_T$, we first project each the word w_t in the corpus to a distributed feature vector in the word embedding layer. The multi-layered LSTM then sequentially takes these embeddings as well as the past hidden state as input and outputs the corresponding context vector. Next, part of the context vector is randomly discarded in the DropContext layer. Finally, the updated context vector and final hidden state of the encoder are fed into the iterative attention decoder, iteratively updates the attentions and refines the current predictions. At the end of the decoder, the fully connected layer with a $softmax$ layer will produce a probability distribution over all the words in the vocabulary.

2.1 Word Embedding

Word embedding is the concept of projecting each word in a vocabulary to a distributed word feature vector. Word embedding plays an important role in language modeling [18]. As pointed out by Bengio et al. [4], word embedding helps a network to fight the curse of dimensionality with distributed representations. Through word embedding, semantically similar words, such as ‘cat’ and ‘dog’, are expected to have a similar embedding feature; thus, a training sample that contains ‘cat’ can easily be projected to the case of ‘dog’ and vice versa. Accordingly, word embedding reduces the number of training samples requirement and, more importantly, alleviates the curse of dimensionality. Additionally, word embedding, i.e., the feature vector of each word, is directly learned from the corpora and is naturally trained with neural networks, such as RNN and LSTM, in an end-to-end manner. Given the advantages of word embedding, we used it for word representation at the bottom of our word association system, as shown in Fig. 1, to be jointly trained with the encoder and iterative attention decoder.

2.2 Iterative Attention Decoder (IAD)

In the previous works, the attention-based decoder only ‘glance’ at the source information once, and may make an inappropriate decision. Therefore, we herein employ an iterative attention decoder to our system, giving us a chance to ‘view’ the source information again and refine the current predictions.

From the multi-layered LSTM encoder, we obtain the source hidden state $\boldsymbol{c_n}$ with a T dimension, which is the same as the number of the input words. Additionally, a current target hidden state $\boldsymbol{h_n}$ is output from the decoder. Therefore, we can formulate the iterative attention decoder as:

$$\boldsymbol{y_n} = \mathrm{IAD}(\boldsymbol{c_n}, \boldsymbol{y_{n-1}}) \tag{2}$$

where $\boldsymbol{y_{n-1}}$ is the last output of the IAD system. Note that, when $n = 1$, $\boldsymbol{y_0}$ is uniform distribution, and Eq. (2) is updated for N times in the form of a recurrent neural network.

Inspired by the work of Luong [12], we attempt to employ a context vector $\boldsymbol{c_n}$ that captures relevant input information to aid in the prediction of $\boldsymbol{y_n}$, and Eq. (2) can be executed in two step:

(1) We calculate the aligned weights $\boldsymbol{\alpha_n}$ according to the source context vector $\boldsymbol{c_n}$ and the current target hidden state $\boldsymbol{h_n}$:

$$\boldsymbol{\alpha_n^s} = \frac{\exp(\boldsymbol{\gamma_n^s})}{\sum_{t=1}^{T} \exp(\boldsymbol{\gamma_n^t})} \tag{3}$$

where s is the dimension index of both $\boldsymbol{\alpha_n}$ and $\boldsymbol{\gamma_n}$. Here, the content-based score $\boldsymbol{\gamma_n^t}$ can be denoted as:

$$\boldsymbol{\gamma_n^t} = \boldsymbol{v_a^\top} \tanh(\boldsymbol{W_a}[\boldsymbol{h_n^\top}; \boldsymbol{c_n^t}]) \tag{4}$$

Note that, both $\boldsymbol{v_a^\top}$ and $\boldsymbol{W_a}$ are learnable parameters and $[\cdot]$ is the concatenation operation. Subsequently, we adopt the soft attention mechanism [19] where the updated context vector uctv_t is defined as the weighted sum of the source context vector.

$$\mathrm{uctv}_t = \sum_{t=1}^{T} \boldsymbol{\alpha_n^t} \boldsymbol{c_n^t} \tag{5}$$

(2) The decoder iteratively updates the attentions and refines the current predictions using a recurrent neural network:

$$\boldsymbol{y_n} = \mathrm{RNN}(\mathrm{uctv}_t, \boldsymbol{y_{n-1}}) \tag{6}$$

where the RNN is implemented by a variant of recurrent neural network: Gated Recurrent Unit (GRU) [20]. Compared with LSTM, GRU only contains two gating units that modulate the flow of information, therefore, costing lower consumption.

In the last time step, the fully connected layer with a $softmax$ layer will produce a probability distribution over all the words in the vocabulary.

2.3 DropContext (DC)

To overcome the over-fitting problem of attention model, we propose DropContext, a new training strategy, to enhance the efficiency of the learning process of attention model, as shown in the black dotted line in Fig. 1.

Suppose that we have the source context vector $\boldsymbol{c_n}$, which is a set of T-dimensional vectors, thus we can update the context vector with DropContext layer:

$$\boldsymbol{c_n^{'}} = \mathrm{DC}(\boldsymbol{c_n}) \tag{7}$$

Many attempts have been performed to execute the DropContext layer in our early work, considering the balance between performance and consumption. Our DropContext layer is implemented in two steps. First, we construct a T-dimensional drop-mask $\mathbf{M}$, which is randomly initialized by the drop-ratio θ:

$$\mathbf{M} = \{m_t = \mathbb{I}\{\zeta > \theta\}, t = 1, 2, \cdots, T\} \tag{8}$$

where $\mathbb{I}\{\cdot\} = 1$ when the condition is true and otherwise zero. It is noteworthy that ζ can follow any distribution, e.g., Gaussian distribution ora exponential distribution. In this paper, ζ follows a uniform distribution.

Subsequently, we update the source context vector by the element-wise product between $\boldsymbol{c_n}$ and $\mathbf{M}$:

$$\boldsymbol{c}_{\boldsymbol{n}}^{'} = \boldsymbol{c_n} \odot \mathbf{M} \tag{9}$$

We have to claim that, after introducing the DropContext layer, we only need to replace $\boldsymbol{c_n}$ with $\boldsymbol{c}_{\boldsymbol{n}}^{'}$ in Eqs. (4) and (5) for the iterative attention decoder.

2.4 Word Association

By integrating the multi-layered LSTM encoder and iterative attention decoder with the prediction layer, from the bottom to the top, we construct a word association system. Formally, the word association system employs the chain rule to model joint probabilities over word sequences:

$$p(w_1, ..., w_N) = \prod_{i=1}^{N} p(w_i | w_1, ..., w_{i-1}) \tag{10}$$

where the context of all the previous words is encoded with LSTM and updated as the predicted word is added. The probability of words is generated through the *Softmax* layer.

The process of associating words of arbitrary length is shown in Fig. 2. Our word association system takes the words of a given sequence as the input. The system then associates the next word by generating a probability distribution over all the given words, as the number upon the black lines shown in Fig. 2. Therefore, we can sort the predicted words in descending order of probability.

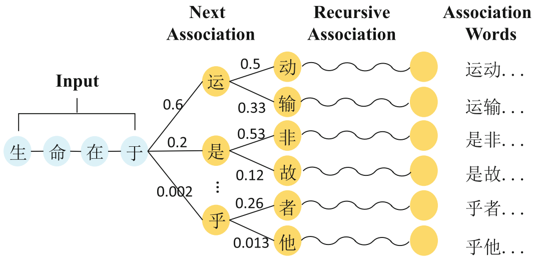


Fig. 2. Schematic diagram of word association. Given the beginning words as input, our word association system predicts a list of candidate words. By recursively adding these candidate words into the input, our word association system can associate sentence of arbitrary length, which is syntactically reasonable. Note that, the numbers upon the black lines represent the probability of the next word.

We adopt the first or top three in the list as the input for the next time step, and associate the following words in the same way. Finally, the system provides candidate associated sentences and their own probability. As described in Fig. 2, after taking the the initial words, our word association system produces a list of candidate words. By associating words in a recursive manner, our word association system manages to generate syntactically reasonable sentences of arbitrary length.

3 Experiments

3.1 Dataset

There is lack of benchmark dataset for the research on word association. Typically, researchers employ their own text corpus to generate the language model. To present an objective evaluation of our word association system, we use two publicly available text corpora, CLDC corpus [21] collected by the Institute of Applied Linguistics, and the Three Hundred Tang Poems (THTP corpus) [22].

For the CLDC corpus, we extracted the available data and filtered extremely rare Chinese characters and characters in other languages. The dataset contains 3455 classes and is divided into two groups, with approximately 70% of data used for training and the remainder for testing. Consequently, the training set contains 59,019,610 words and the test set contains 25,294,119 words.

The THTP corpus consists of 310 poems written by 77 famous poets during the Tang dynasty. For convenience, the punctuation has been removed from the poems. The dataset has approximately 20,000 words and consists of 2,497 classes, including a special symbol that indicates the end of a sentence.

3.2 Implementation Details

The proposed multi-layered LSTM encoder consists of two layers with the hidden size of 512, which are unrolled for 10 steps. Additionally, we also use dropout with probability 0.5 for our LSTMs. Besides, the iterative attention decoder is implemented with an attention-based GRU, whose hidden size is 512. To strike a balance between performance and consumption, we set the maximum iteration N as 3 for the little performance gain with larger N. We train the system in an end-to-end manner using stochastic gradient descent with a weight decay of 0.0005, momentum of 0.9, and gradient clipping set to 10. The initial learning rate is set to 0.1, followed by a polynomial decay of power 0.5.

In this paper, we use the canonical performance metric of language models, namely the perplexity [23], to evaluate our word association system. Perplexity measures the average number of branches of the predicted text, the reciprocal of which can be seen as the average probability of each word. Formally, perplexity is calculated as:

$$\text{perplexity} = \sqrt[K]{\frac{1}{e^{(-\sum \log(\boldsymbol{p}(w)))}}} \tag{11}$$

where $\boldsymbol{p}(w)$ is the probability of each word in the test set and K is the total number of words that appeared in the test set. It is noteworthy that the word association system with a low perplexity generally performs better than those with a higher perplexity. Besides, we also perform many visualizations of the experiment result, which are more obvious.

3.3 Effectiveness of the DropContext Layer

In this section, we perform a detailed analysis on the performance of our proposed DropContext method. In Table 1, we compare the performance of the system with different drop-ratios. When the drop-ratio is 0.0, no DropContext is available in our model and it is set as the baseline in our experiments. As the drop-ratio increases, the gap between train loss and test loss became smaller, and the system performance improves, i.e., the perplexity and testing loss of the system decreases. We can conclude that, by introducing the DropContext, the over-fitting during the training procedure can be alleviated. However, the system performance decreases afterward when the drop-ratio is lager than 0.4. This is because when the drop-ratio is too large, too much context information will be discarded in the training procedure, which will confuse the decoder and render our system difficult to converge.

Table 1. Influence of drop-ratio

Drop-ratio	0.0 (baseline)	0.2	0.4	0.6	0.8
Train loss	2.63	4.13	**4.37**	4.45	4.89
Test loss	4.79	3.92	**3.86**	3.89	4.42
Perplexity	120.36	50.40	**47.46**	48.91	83.10

3.4 Effectiveness of the Iterative Attention Decoder

In this section, we compare the proposed iterative attention model with a regular LSTM-based model similar to that reported by Merity et al. [5]. The regular LSTM-based model consists of two LSTM layers, with the hidden size of 512, which is the same as the multi-layer LSTM encoder in our system. The difference between the regular LSTM-based model and our model is that each hidden state of the former is followed by the fully connected layer and a softmax layer. This means that once a word is input, the system can only make a 'decision' (prediction) once. Note that, both of them are trained with the CLDC corpus.

As shown in Table 2, the regular LSTM-based model (denoted as R-LSTM) achieves a perplexity of 62.80. By introducing the iterative attention decoder, our model (denoted as IA-LSTM) achieves a much lower perplexity of 47.46. We can conclude that adding iterative attention mechanism can lead to a better performance.

Table 2. Perplexity and test loss on the CLDC corpus

Method	Perpelxity	Test loss
R-LSTM [5]	62.80	4.14
IA-LSTM	47.46	3.86

Additionally, Fig. 3 shows several examples on how the proposed iterative attention decoder iteratively updates the attentions and refines the current predictions. As we can see, although the model may make an inexact prediction at the beginning, it can update the attentions to focus on the last few words and make a more reasonable prediction. This is also corresponds to common sense that the associated words are more related with their adjacent words [24].

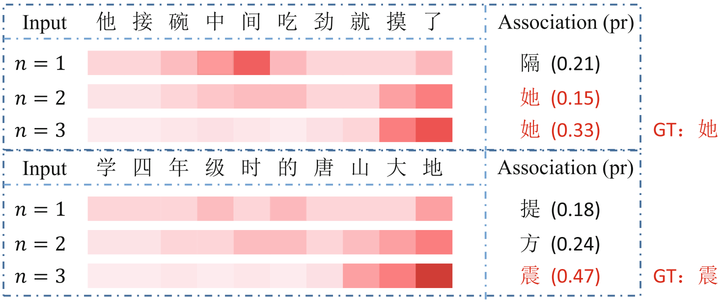
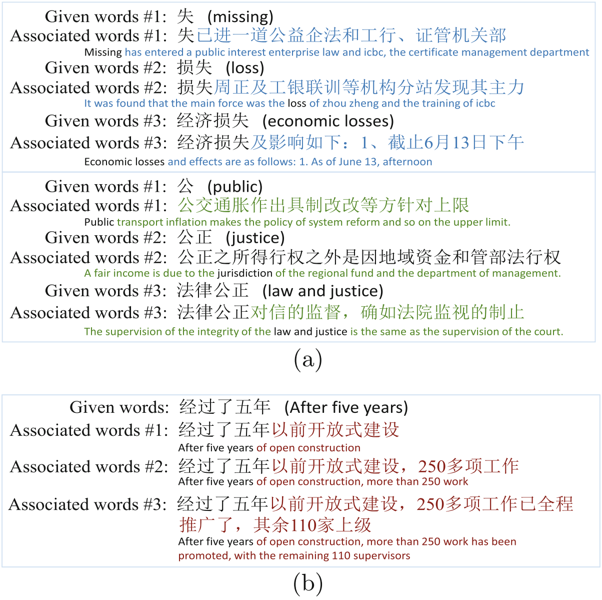


Fig. 3. Examples on how the proposed iterative attention decoder iteratively updates attentions and refines current predictions. At each time-step n, the current association word is listed. Each result is followed by the corresponding probability. Words in red are the most appropriate ones. Note that we use red squares to display the attention weight of each word, the deeper the color is, the greater the weight is.

3.5 Output Visualization of Word Association System

Our word association system generates an arbitrary length string of associated words. The more information is provided to the system, the more meaningful words will be generated. As shown in Fig. 4(a), given different numbers of words as beginning, our system associates sentences with completely different meanings. When only less information is available, the system randomly generates the sentences. However, when given more detailed information, the system associates a sentence that is quite relevant to the given words. In Fig. 4(b), the words in the first line are the input to the word association system and the subsequent lines are the associated sentences of different lengths. Note that regardless of the length of the associated sentences, they are reasonable and meaningful.



Given words #1: 失 (missing)
Associated words #1: 失已进一道公益企法和工行、证管机关部
Missing has entered a public interest enterprise law and icbc, the certificate management department
Given words #2: 损失 (loss)
Associated words #2: 损失周正及工银联训等机构分站发现其主力
It was found that the main force was the loss of zhou zheng and the training of icbc
Given words #3: 经济损失 (economic losses)
Associated words #3: 经济损失及影响如下：1、截止6月13日下午
Economic losses and effects are as follows: 1. As of June 13, afternoon

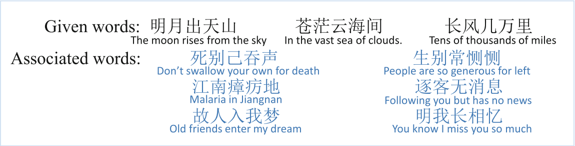
Given words #1: 公 (public)
Associated words #1: 公交通胀作出具制改改等方针对上限
Public transport inflation makes the policy of system reform and so on the upper limit.
Given words #2: 公正 (justice)
Associated words #2: 公正之所得行权之外是因地域资金和管部法行权
A fair income is due to the jurisdiction of the regional fund and the department of management.
Given words #3: 法律公正 (law and justice)
Associated words #3: 法律公正对信的监督，确如法院监视的制止
The supervision of the integrity of the law and justice is the same as the supervision of the court.

(a)

Given words: 经过了五年 (After five years)
Associated words #1: 经过了五年以前开放式建设
After five years of open construction
Associated words #2: 经过了五年以前开放式建设，250多项工作
After five years of open construction, more than 250 work
Associated words #3: 经过了五年以前开放式建设，250多项工作已全程推广了，其余110家上级
After five years of open construction, more than 250 work has been promoted, with the remaining 110 supervisors

(b)

Fig. 4. Output of word association system. In (a), there are three kinds of inputs to the system, ordered by the amount of information in Chinese. In (b), there are three different lengths of output for the same input to the system. The associated sentence is syntactically reasonable for any arbitrary length. The tiny English sentence right below the Chinese sentence is the corresponding translation.



Given words: 明月出天山 苍茫云海间 长风几万里
The moon rises from the sky In the vast sea of clouds. Tens of thousands of miles

Associated words:
死别已吞声
Don't swallow your own for death
江南瘴疠地
Malaria in Jiangnan
故人入我梦
Old friends enter my dream

生别常恻恻
People are so generous for left
逐客无消息
Following you but has no news
明我长相忆
You know I miss you so much

Fig. 5. Result of the model trained with the THTP corpus (shown in poetry format). Given arbitrary words, our system associates a meaningful poem with the Tang poem style.

3.6 Generating Poems

To verify the significance of our word association system, an poetry generating experiment is conducted using the THTP corpus. In the testing phase, a contiguous piece of a sentence is input to the word association system, and the system attempts to associate a poem accordingly.

To generate a poem, as shown in Fig. 5, arbitrary words are given to the association system. Staring with the given words, the system produces a meaningful poem of the Tang poem style. Furthermore, the associated poem is incredibly 'real' that it is difficult to distinguish whether it is one of the original poems in the dataset.

4 Conclusion

In this paper, we presented a flexible Chinese word association method which consists of a multi-layer LSTM encoder and an iterative attention decoder. Experiments show that the attention mechanism can improve the performance of Chinese word association system. Besides, the iterative attention decoder implemented in our system can iteratively uses its previous prediction to update attentions and to refine current predictions. Moreover, by adopting the DropContext layer in our proposed model, over-fitting can be avoided during the training procedure, which is proved to be better converged. Additionally, we showed that our system can generate syntactically reasonable associated words of arbitrary length and tends to associate more meaningful yet relative words when given more context information. Finally, we verify the significance of our word association system through an interesting poem generating experiment.

Acknowledgement. This research is supported in part by GD-NSF (no. 2017A030312006), the National Key Research and Development Program of China (No. 2016YFB1001405), NSFC (Grant No.: 61673182, 61771199), and GDSTP (Grant No.: 2014A010103012, 2017A010101027), GZSTP(no. 201607010227).

References

1. Swiffin, A.L., Pickering, J.A., Arnott, J.L., Newell A.F.: PAL: an effort efficient portable communication aid and keyboard emulator. In: ACRT, pp. 197–199 (1985)
2. Carlberger, A., Carlberger, J., Magnuson, T., Hunnicutt, M.S., Palazuelos-Cagigas, S.E., Navarro, S.A.: Profet, a new generation of word prediction: an evaluation study. In: Proceedings, ACL Workshop on Natural Language Processing for Communication Aids, pp. 23–28 (1997)
3. Shein, F., Nantais, T., Nishiyama, R., Tam, C., Marshall, P.: Word cueing for persons with writing difficulties: WORDQ. In: Proceedings of CSUN 16th Annual Conference on Technology for Persons with Disabilities (2001)
4. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(Feb), 1137–1155 (2003)
5. Merity, S., Keskar, N.S., Socher, R.: Regularizing and optimizing LSTM language models. CoRR, abs/1708.02182 (2017)
6. Yang, Z., Dai, Z., Salakhutdinov, R., Cohen, W.W.: Breaking the softmax bottleneck: a high-rank RNN language model. In: ICLR (2018)
7. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH, vol. 2, pp. 3 (2010)
8. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: ICML, pp. 1017–1024 (2011)
9. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: EMNLP, pp. 670–680 (2014)
10. Hinton, G., Deng, L.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
11. Jenckel, M., Bukhari, S.S., Dengel, A.: Training LSTM-RNN with imperfect transcription: limitations and outcomes. In: Proceedings of the 4th International Workshop on Historical Document Imaging and Processing, pp. 48–53. ACM (2017)

12. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025 (2015)
13. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: NIPS, pp. 577–585 (2015)
14. Dzmitry B., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
15. Yang, W., Jin, L., Tao, D., Xie, Z., Feng, Z.: DropSample: a new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition. Pattern Recognit. **58**, 190–203 (2016)
16. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors 3 July 2012. CoRR, abs/1207.0580 (2016)
17. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using DropConnect. In: ICML (2013)
18. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: ACL, pp. 310–318. ACL (1996)
19. Show, attend and tell: neural image caption generation with visual attention. In: ICML (2015)
20. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078 (2014)
21. Chinese linguistic data consortium. The Contemporary Corpus developed by State Language Commission P. R. China, Institute of Applied Linguistics (2009). http://www.chineseldc.org. Accessed 22 Oct 2016
22. Wikipedia. Three Hundred Tang Poems (2018). https://en.wikipedia.org/wiki/Three_Hundred_Tang_Poems
23. Jurafsky, D., James, H.: Speech and language processing an introduction to natural language processing, computational linguistics, and speech (2000)
24. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based N-gram models of natural language. Comput. Linguist. **18**, 467–479 (1992)