A LSTM Approach with Sub-word Embeddings for Mongolian Phrase Break Prediction

Rui Liu, Feilong Bao ✉, Guanglai Gao, Hui Zhang, Yonghe Wang
College of Computer Science, Inner Mongolia University,
Inner Mongolia Key Laboratory of Mongolian Information Processing Technology,
Hohhot 010021, China
liurui_imu@163.com;csfeilong@imu.edu.cn

Abstract

In this paper, we first utilize the word embedding that focuses on sub-word units to the Mongolian Phrase Break (PB) prediction task by using Long Short-Term Memory (LSTM) model. Mongolian is an agglutinative language. Each root can be followed by several suffixes to form probably millions of words, but the existing Mongolian corpus is not enough to build a robust entire word embedding, thus it suffers a serious data sparse problem and brings a great difficulty for Mongolian PB prediction. To solve this problem, we look at sub-word units in Mongolian word, and encode their information to a meaningful representation, then fed it to LSTM to decode the best corresponding PB label. Experimental results show that the proposed model significantly outperforms traditional CRF model using manually features and obtains 7.49% F-Measure gain.

1 Introduction

A Text-to-Speech (TTS) system converts the input text into synthetic speech with high naturalness and intelligibility. Naturalness is mainly influenced by the prosody modeling, especially by the Phrase Break (PB) prediction. Because the PB prediction is the first step of TTS, any error in this step will propagate to downstream steps such as intonation prediction and duration modeling. Those errors will result in the synthetic speech which is unnatural and difficult to understand. So that many researchers devote themselves to improving the performance of the PB prediction.

Typically PB prediction methods usually use machine learning models like Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs) which trained with large sets of labeled training data. In these PB prediction models, the Part-of-Speech (POS) tag have been shown to be an effective feature and usually been included in the input feature set. The POS estimation itself is also a challenging task, and relies on large labeled training corpus, too. Its accuracy is always lower than our expectation, especially for those low-resource languages like Mongolian where the required linguistic resources are not readily available, and manual annotation is expensive and time-consuming.

In recent years, there are many works applying the word embedding techniques to Natural Language Processing (NLP) tasks, such as question answering, machine translation and so on (Bordes, 2014; Xiong, 2017; Devlin, 2014). Previous work has shown that the POS prediction task can be solved with high accuracy only using the word embedding feature as the input (Wang, 2015). POS information is most likely to be included in the word embedding representations. Therefore, some PB prediction systems which don't rely on the POS feature are developed (Watts, 2011; Vadapalli, 2014; Vadapalli, 2016). In (Watts, 2011), the authors obtain continuous-valued word embedding features that summarize the distributional characteristics of word types as surrogates of POS features. In (Vadapalli, 2014), researchers propose a neural network dictionary learning architecture to induce task-specified word embedding representations and show that these features perform better at PB prediction task. (Vadapalli, 2016) presents their investigations of recurrent neural networks (RNNs) for the phrase break prediction task by using word embedding. The above efforts have also been directed toward unsupervised methods of inducing word representations, which can be used as surrogates for POS tags, in the PB prediction task.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2448–2455
Santa Fe, New Mexico, USA, August 20-26, 2018.

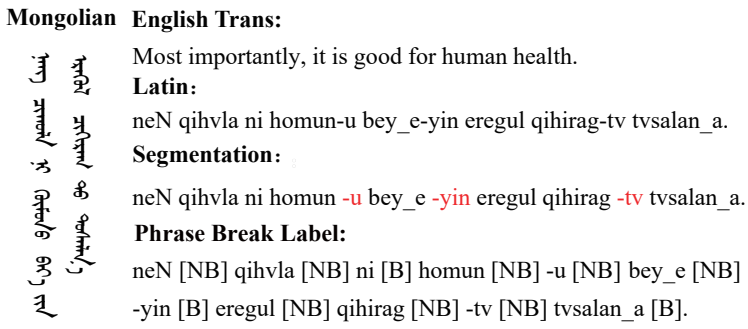

Figure 1: NNBS suffixes within a Mongolian sentences, the red part is the segmented NNBS suffixes from the word. There are three pauses in the sentence, one of which is located at the NNBS suffix: "-yin".

Although the word embedding training operates in an unsupervised way, this approach face an issue when applied to Mongolian languages, which are agglutinative in nature and the available Mongolian corpus is not large enough for the huge Mongolian vocabulary. Fortunately, Mongolian is a morphologically rich language. Its suffixes often act as a positive signal which implies the POS information of the word. It's like that the word implied by the suffix '-ly' is an adverb in English. Morphologically, unlike many other languages, a Mongolian word is not just a concatenation of characters. It is constructed by the special agglutinative property. Mongolian words can be decomposed into a set of morphemes: one root and several suffixes.

In this paper, we investigate Mongolian PB prediction models that operate on the level of sub-word units: stem and suffixes (the part without suffix). We hypothesize that stem and suffix serve to discriminate words based on syntactic meaning, and that these sub-word units can be used to model PB. We automatically segment every Mongolian word to a sequence of sub-word units, then map all sub-word units into a continuous vector representations by lookup table, which are then fed into a neural network. Instead of a feed-forward network, we use the Long Short-Term Memory (LSTM) network to predict the right PB label. The segmentation process reduces the vocabulary and alleviates the data sparse problem. Therefore, the learned word embedding for sub-word is more robust, then the performance of the PB prediction system can be improved.

Our experiments show the proposed model can achieve significant performance than the conventional CRF-based models, and the sub-word embedding based method outperforms the entire word embedding based method.

2 Mongolian characteristics

As an agglutinative language, like Turkish, Japanese and Korean, Mongolian has complex morphological structure. Most Mongolian words can be decomposed into root, derivational suffixes, and inflectional suffixes (Bao, 2013). The first two parts together are called a word-stem, it holds the major information in a word, and inflectional suffixes server to discriminate words based on lexical meaning. As for nouns, the inflectional suffixes contain case suffixes, reflexive suffixes, and plural suffixes. All above three suffixes are attached to stem through a Narrow Non-Break Space (NNBS) (U+202F, Latin: "-"). We call them NNBS suffixes. The NNBS suffixes used are very pervasive, in Fig.1, there are 3 NNBS suffixes in a sentence with only 8 words.

New words can be formed by connecting different suffixes to the end of a stem. A lot of new words can be induced from a single stem. For example:  ,  ,  ,  ,  . These words share the same word-stem "" (Latin: "sandali", means: "chair"). It makes Mongolian has a huge vocabulary, about one million, with only about 30 thousands stems (Bao, 2013). The large vocabulary leads to data sparse, and a serious dependence on a large corpus. However, the available Mongolian corpus is not enough for the word embedding training. The bad word embedding further reduces the accuracy of the PB prediction system. To get through the problem, we segment the NNBS suffixes from the Mongolian

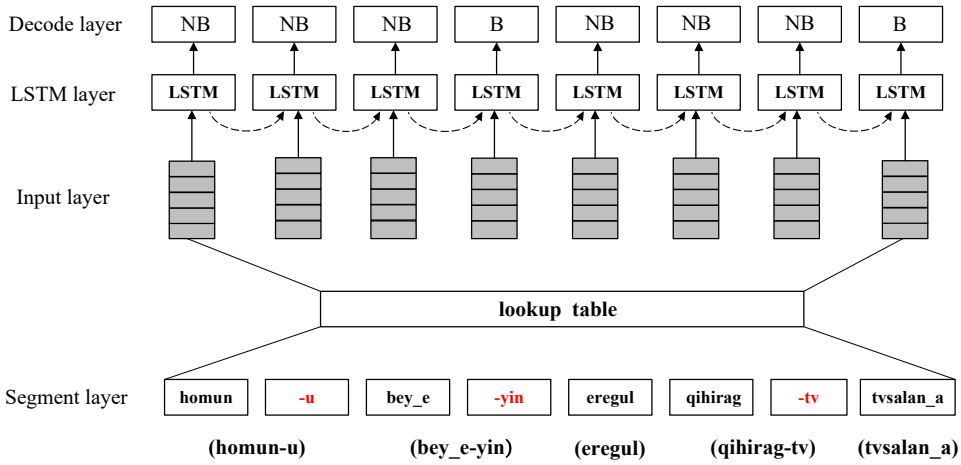

Figure 2: A framework for the proposed Mongolian PB prediction system based on sub-word embedding. The segment layer convert the Latin-cased Mongolian word to its sub-word form: stem (black part) and suffixes (red part); the sub-word embeddings are generated by a lookup table; the information is passed through a LSTM layer and the decode layer.

word and learn embedding representation on these individual suffixes and stems. After segmentating, the sentence will include more tokens, for example, in Fig.1, 3 words with NNBS will be turned into new units: "*homun-u*" turned into "*homun*" and "*-u*", "*bey_e-yin*" turned into "*bey_e*" and "*-yin*", "*qihirag-tv*" turned into "*qihirag*" and "*-tv*".

3 Proposed model

Our system framework is shown in Fig.2. This architecture consists of a segment layer, an input layer, a LSTM layer and a decode layer.

The system input is raw Mongolian sentence consisting of entire word. First, the segment layer converts every Mongolian word into stem and NNBS suffixes according to the location of NNBS inside of the word. Second, the input layer maps these processed Mongolian sub-word units into sub-word embeddings. The remaining layers are a LSTM network (Greff, 2017) used as a discriminative classifier and a decode layer to obtain the final PB label: "B" or "NB". "B" and "NB" are PB labels means *break after a word* and *non-break* respectively.

3.1 Sub-word embedding

In current work, we hypothesize that stem and suffix serve to discriminate words based on semantic meaning in Mongolian, and that these sub-word units can be used to model PB. We learn the embedding representation for sub-word units inspired the word embedding technique.

Word embedding represents words as continuous vectors in a low-dimensional space based on the distributional hypothesis that words in similar contexts have an analogous meaning. Based on this hypothesis, various word embedding models have been developed, including continuous bag-of-words model (CBOW), Skip-Gram model (Mikolov, 2013), and Global C&W(Glove) (Pennington, 2014). We use Skip-Gram model to train the sub-word embedding representation. Given a sequence of training unit $u_1, ..., u_T$, the Skip-Gram model try to maximize the average of log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c}\log P\left(u_{t+c}|u_t\right) \tag{1}$$

where c is the training context around the center unit u_t. The prediction probability can be defined as:

$$P(o|i) = \frac{\exp(V_o^{\mathrm{T}}W_i)}{\sum_{u=1}^{U}\exp(V_u^{\mathrm{T}}W_i)} \tag{2}$$

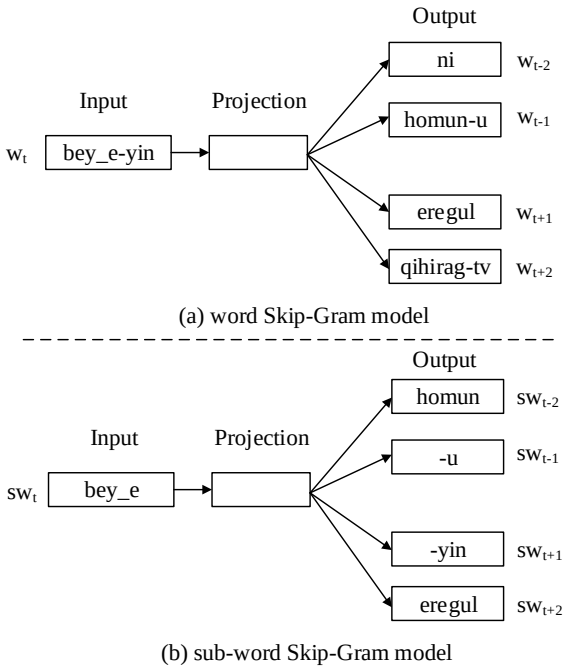

Figure 3: Comparision of the Mongolian Skip-Gram model representation of word unit (a) and sub-word unit (b). The sub-word Skip-Gram model regard stem and suffix as a basic unit, it allows the model to learn more information from suffixes with same context windows.

where W and V are the "input" and "output" unit vector representations of u, and U is the set of sub-word units. X^{T} is the transpose of matrix X.

The difference between word and sub-word Skip-Gram model is shown in in Fig.3 for the sentence: " *homun-u bey_e-yin eregul qihirag-tv tvsalan_a*". In the word Skip-Gram model, if the center word is "*bey_e-yin*", the nearby words are "*ni*", "*homun-u*", "*eregul*" and "*qihirag-tv*". While in sub-word Skip-Gram model, when the basic learning unit is changed to sub-word, the center word is turned to "*bey_e*", the nearby words to "*homun*", "*-u*", "*-yin*" and "*eregul*". The sub-word Skip-Gram model lies in dealing with sub-word units (stem and suffixes). It captures more information from the nearby suffixes under same context window size.

3.2 LSTM layer & Decode layer

PB prediction can be treated as a sequential labeling task that assigns boundary labels to words of an input sentence. Recurrent neural networks (RNNs) have recently produced outstanding performances on many tasks including sequential labelling (Vadapalli, 2016). In theory, RNN can learn from the entire historical inputs. But in practice, it can access only a limited range of context because of the vanishing gradient problem. LSTM uses purpose-built memory cells to store information, which is designed to overcome this problem. LSTM is composed of a set of recurrently connected memory blocks and each block consists of one or more self-connected memory cells and three multiplicative gates, i.e., input gate, forget gate and output gate. The three gates are designed to capture long-range contextual information by using nonlinear summation units.

Specifically, in this study, we follow the LSTM with forget gates and peephole connections to predict Mongolian phrase break. We use the following implementation:

$$i_t = \sigma(W_{xi}x_t) + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3}$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where $\odot$ indicates element-wise product and σ indicates element-wise sigmoid function. More detail description can be found in (Greff, 2017).

The LSTM is trained using standard backpropagation through time to maximize the data conditional likelihood:

$$\prod_t P(y_t|x_1 \cdots x_t) \tag{7}$$

where x_t, y_t are the input and output respectively at time t. As mentioned in Section 3.1, the input x_t to the LSTM at time t, is the sub-word embedding corresponding to the token at time t.

The probability distribution is strictly a function of the hidden layer activations, which in turn depend only on the inputs (and their own past values). Thus, the most likely sequence of phrase break labels can be computed as:

$$y_t^* = \arg\max P(y_t|x_1 \cdots x_t) \tag{8}$$

4 EXPERIMENTS AND RESULTS

4.1 Dataset

4.1.1 Embedding corpus

The embedding train data were crawled from mainstream websites in Mongolian. After cleaning web page tags and filtering longer sentences, its token size and vocabulary are about 200 million and 3 million respectively. After we split the suffixes into a new token, we find a dramatic decrease on vocabulary even with the token number growth. The segmented corpus' token size and vocabulary are about 300 million and 2.5 million respectively.

4.1.2 PB prediction corpus

To evaluate the proposed method, a Mongolian speech synthesis corpus is involved. This corpus is recorded by a native Mongolian female speaker, who is a television news announcer. This corpus contains 58,695 utterances, where there are about 449,000 Mongolian words and 22,050 vocabularies. This corpus is labeled with prosodic phrase boundary by hand according to its speech. The labels correspond to the actual stops in the utterance. Specifically, each word is labeled as "B" (means *break*) if there is a short break after the word. Otherwise, the word is labeled as "NB" (means *non-break*). We divide the corpus into suitable subsets for training, validation and test as 8:1:1.

4.2 Evaluation

We conduct three sessions of experimentation. The first session is designed to verify the effectiveness of sub-word method under the CRF model, which aims to investigate the performance of regard the stem and suffixes as individual tokens. The second session about CRF-based systems is built to evaluate the idea of replacing the POS with the embedding representation for word and sub-word. The third session aims to investigate whether utilizing the LSTM model with sub-word embedding can improve the performance of Mongolian PB prediction. All Mongolian words are Latin-cased before passing through the lookup table to convert to their corresponding embeddings.

We evaluate the systems with F-Measure of the PB prediction, which this the harmonic mean of the *Precision* and the *Recall*. F-Measure values range from 0 to 1. The higher F-Measure means better PB prediction performance.

4.2.1 CRF with subword

We use CRF++ toolkit[1] to build a CRF-based Mongolian phrase break prediction system as a baseline named "CPw". "CPw" system consider the nearby words and its POS feature within the fixed context window size. Another two CRF-based systems are built to analyze the effects of the sub-word named "CPs" and "CPB", which have the same configuration as the CRF baseline, except the word feature. The 'CPs' system removes the segmented suffixes from the input token. The 'CPB' system segments the

[1] *https://taku910.github.io/crfpp/*

word into stem and suffixes, and uses them both as individual input tokens. All the CRF models used in this paper is a linear-chain CRF, we carry on all experiments under two context windows type: Unigram (U: previous one word, current word and future one word) and Bigram (B: previous two words, current words and future two words).

As illustrated in Table 1. Compared with the 'CPw' baseline under all context windows type, we can see the performance of the 'CPs' drop down. It is because that the word stems have less discriminative information than the entire word, since its suffixes are removed. 'CPB' outperforms the baseline and achieves the peak performance (82.96%) under Bigram context window. It can learn all semantic information from the individual suffix and stem but also alleviate the data sparse problem in a limited corpus.

Model	**F-Measure (U)**	**F-Measure(B)**
CPw	82.23	82.40
CPs	82.12	82.34
CPB	82.52	**82.96**

Table 1: Performance of F-Measure for CRF-based model with different context window types. (U: Unigram, B: Bigram)

	F-Measure (U)					**F-Measure (B)**				
Dim / **Model**	50	100	150	200	300	50	100	150	200	300
CEw	82.67	82.89	82.85	82.81	82.73	82.86	82.92	82.98	82.87	82.78
CEs	82.67	82.70	82.73	82.68	82.59	82.65	82.73	82.83	82.80	82.79
CEB	83.45	83.59	83.68	83.44	83.53	83.66	83.72	**83.79**	83.68	83.55

Table 2: Performance of F-Measure for CRF-based model using embeddings for word and sub-word with different dimensions and different context window types. (U: Unigram, B: Bigram, Dim: embedding dimension)

	F-Measure				
Dim / **Model**	50	100	150	200	300
LEw (Vadapalli, 2016)	85.63	85.74	85.89	85.78	85.64
LEs	84.78	84.83	85.01	84.88	84.78
LEB **(proposed)**	89.51	89.77	**89.89**	89.79	88.93

Table 3: Performance of F-Measure for LSTM-based model using embeddings for word and sub-word with different dimensions. (Dim: embedding dimension)

4.2.2 CRF with sub-word embedding

In these systems, the POS feature is replaced by the embedding, i.e. the systems input is the word or sub-word unit and its corresponding embedding. And follow the experimental setting of the Section 4.2.1, we index the three system as 'CEw', 'CEs' and 'CEB' respectively, which denotes CRF model using embedding feature on the entire word, word stem or both the suffix and stem. We test the performance of all systems with five embeddings dimension: 50, 100, 150, 200, 300. The evaluation results are listed in Table 2.

Comparing these three systems, we get the same conclusion as the previous experiment, sub-word based methods performance is better than the entire word setting. For Bigram context, 'CEB' achieves the highest F-Measure (83.79%) in all embedding dimensions. The performance of all systems is first

raised and then decreased with the embedding dimension range from 50 to 300, and reaches the best in 150. While a too long dimension will include other boring information that decoder cannot utilize, a too small dimension can not learn enough information from context. More informative, the performance of the Unigram context is worse than that of the Bigram context but shows the same trend.

The systems using sub-word embedding feature performs better than the systems utilizing POS feature (Section 4.2.1). As can be seen, the performances of 'CEw', 'CEs', 'CEB' are obviously higher than that of 'CPw', 'CPs', 'CPB'. This is mainly caused by the representation power of the word embedding technique. By using the sub-word embedding, we make a better use of the very limited training data in Mongolian. And again the proposed sub-word method alleviate the data sparse problem for both the word embedding training and the PB prediction models training.

4.2.3 LSTM with sub-word embedding

In this experiment, we replace the CRF model with the powerful LSTM model. All of the LSTM models used a single hidden layer of 512 units. All models are trained with a momentum of 0.3, an initial learning rate of 0.01. We select *tanh()* as our activation function, the minibatch size and forget gate bias is 1, weight and inner cells initialization are glorot uniform (Glorot, 2010) and orthogonal (Saxe, 2013) respectively. We train the LSTM model 50 epochs according to the development set. The evaluation results are listed in Table 3 under the name of 'LEw', 'LEs' and 'LEB', which means LSTM model using word embedding feature on the entire word (Vadapalli, 2016), sub-word embedding feature on word stem or both the suffix and stem.

Compared with CRF-based systems (Section 4.2.2) the LSTM-based systems show a clear advantage. 'LEw', 'LEs', 'LEB' systems respectively increased the performance by 2.91%, 2.18%, and 6.1% compared with 'CEw', 'CEs', 'CEB' under the optimum embedding dimension – 150. Our proposed method ('LEB') obtains 7.49% F-Measure gain compared with the baseline system - 'CPw' (Section 4.2.1) under 150 embedding dimensions. It is another evidence of the power of the LSTM model. LSTM model is more suitable than the CRF model in the PB prediction task. It shows that the LSTM model can fully absorb the nutrients of the embedding representation and get more benefits from the sub-word embedding.

5 CONCLUSIONS

In this paper, we look at sub-word units and explore the use of sub-word embedding on stem and suffixes to model Mongolian phrase break by using LSTM network. Embedding representation for the sub-word unit is learned in an unsupervised manner from an untagged Mongolian text corpus. These sub-word units can be directly identified from the text with a simple and effective approach. It provides more information for model Mongolian PB and eliminates the need for additional manually features like part-of-speech (POS) taggers. Experimental results demonstrate by means of the objective measure that LSTM model built using these sub-word embeddings perform significantly improvement than conventional CRF model built using POS sequence information. This work can also inspire other agglutinative language research.

Acknowledgements

This research was supports by the China national natural science foundation (No.61563040, No.61773224), Inner Mongolian nature science foundation (No. 2016ZD06) and the Enhancing Comprehensive Strength Foundation of Inner Mongolia University (No. 10000-16010109-23).

References

Lafferty, John and McCallum, Andrew and Pereira, Fernando CN. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Qian, Yao and Wu, Zhizheng and Ma, Xuezhe and Soong, Frank. 2010. Automatic prosody prediction and detection with Conditional Random Field (CRF) models. *Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*, 135–138.

Bordes, Antoine and Chopra, Sumit and Weston, Jason. 2014. Assigning phrase breaks from part-of-speech sequences. *arXiv preprint arXiv:1406.3676*.

Xiong, Caiming and Zhong, Victor and Socher, Richard. 2017. Dynamic coattention networks for question answering. *ICRL*.

Devlin, Jacob and Zbib, Rabih and Huang, Zhongqiang and Lamar, Thomas and Schwartz, Richard M and Makhoul, John. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. *ACL*, 1370–1380.

Wang, Peilu and Qian, Yao and Soong, Frank K and He, Lei and Zhao, Hai. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*.

Watts, Oliver and Yamagishi, Junichi and King, Simon. 2011. Unsupervised continuous-valued word features for phrase-break prediction without a part-of-speech tagger. *Twelfth Annual Conference of the International Speech Communication Association*.

Vadapalli, Anandaswarup and Prahallad, Kishore. 2014. Learning continuous-valued word representations for phrase break prediction. *Fifteenth Annual Conference of the International Speech Communication Association*.

Liu, Rui and Bao, Feilong and Gao , Guanglai and Zhang,Hongwei. 2015. Approach to Predicition Mongolian Prosody Phrase Based on CRF Model. *Proceedings of the National Conference on Man-Machine Speech Communication*.

Bao, Feilong and Gao, Guanglai and Yan, Xueliang and Wang, Weihua. 2013. Segmentation-based Mongolian LVCSR approach. *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8136–8139.

Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Saxe, Andrew M and McClelland, James L and Ganguli, Surya. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Vadapalli, Anandaswarup and Gangashetty, Suryakanth V. 2016. An Investigation of Recurrent Neural Network Architectures Using Word Embeddings for Phrase Break Prediction.. *Interspeech*, 2308–2312.

Pennington, Jeffrey and Socher, Richard and Manning, Christopher. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

S.Loglo and HuaShabao and Sarula. 2010. Research on Mongolian Lexical Parser Algorithm Based on FNA. *Proceedings of the The Second National Symposium on Multi-lingual Knowledge Base Construction*.

Glorot, Xavier and Bengio, Yoshua. 2010. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Greff, Klaus and Srivastava, Rupesh K and Koutník, Jan and Steunebrink, Bas R and Schmidhuber, Jürgen. 2017. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*.