Mongolian Questions Classification Based on Mulit-Head Attention

Guangyi Wang, Feilong Bao, Weihua Wang*
College of Computer Science, Inner Mongolia University, China
Inner Mongolian Key Laboratory of Mongolian
Information Processing Technology, China
wanggycs@163.com
{csfeilong, wangwh}@imu.edu.cn

Abstract

Question classification is a crucial subtask in question answering system. Mongolian is a kind of few resource language. It lacks public labeled corpus. And the complex morphological structure of Mongolian vocabulary makes the data-sparse problem. This paper proposes a classification model, which combines the Bi-LSTM model with the Multi-Head Attention mechanism. The Multi-Head Attention mechanism extracts relevant information from different dimensions and representation subspace. According to the characteristics of Mongolian word-formation, this paper introduces Mongolian morphemes representation in the embedding layer. Morpheme vector focuses on the semantics of the Mongolian word. In this paper, character vector and morpheme vector are concatenated to get word vector, which sends to the Bi-LSTM getting context representation. Finally, the Multi-Head Attention obtains global information for classification. The model experimented on the Mongolian corpus. Experimental results show that our proposed model significantly outperforms baseline systems.

1 Introduction

When people read a specific sentence on a flyer or some magazine, they can understand the context or intent of the sentence. And they can also extract information from the sentence. How to make a computer think like a human. Natural Language Processing (NLP) and Natural Language Understanding (NLU) study how to make the computer understand the semantics of natural language. The computer uses natural language to communicate with people to realize human-machine interaction. Deep learning models have achieved state-of-the-art performance in various natural language processing tasks such as text summarization (Rush et al., 2015), question answering (He and Golub, 2016) and machine translation (Kudo, 2018). In recent years, question answering is a key technology in intelligent applications. It has aroused widespread concern. Pipeline the first task of question system is to classify the domain of the dialogue after the user enters the message (text or voice). Question classification divides questions into several semantic categories. The machine gets a predicted category of the dialogue and the system returns a concise and accurate answer. The understanding of questions provides constraints for improving the accuracy of question answering system. Moldovan et al. (2003) have studied the influence of each part of the question answering system on the system performance. The question classification recognition has the greatest influence on the system performance. Therefore, to get a good question answering system, it is necessary to design a high accuracy model of question classification.

However, the research of the Mongolian questions classification is very fewer. The reason is that Mongolian corpus is scarce and there is no public Mongolian corpus. Data collected from internet are noisy and uncertain in terms of coding and spelling. The word-formation is different from Chinese and English. It consists of roots, stems and affixes. These problems result in unlimited vocabulary. The existing short text classification methods are not effective. How to classify the questions accurately is a complicated problem.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1027-1035, Hainan, China, October 31 - Novermber 1, 2020.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

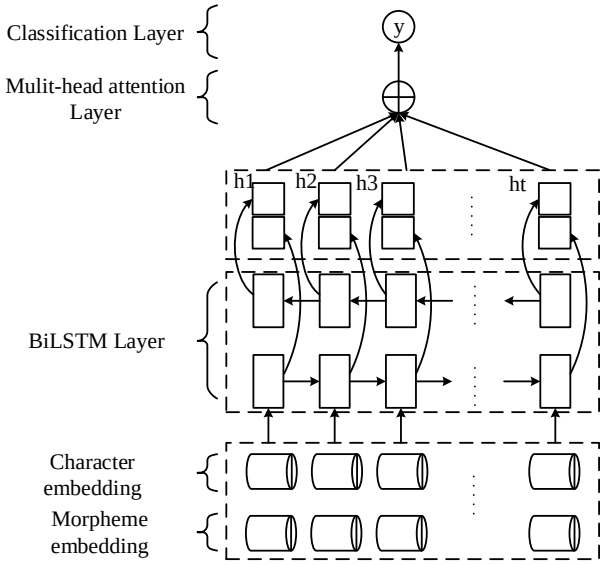


Figure 1: The model architecture of MA-B.

In this article, the training data were crawled from the Mongolian web sites. After cleaning the invalid data, we constructed a question classification data set. We propose a method of the Mongolian question classification, which combines the Bi-LSTM model with the Multi-Head Attention mechanism. As shown in Figure 1, the model is named MA-B. To better learn semantic information from sentences, we introduce the morphemes representation. The character vector and the morpheme vector are concatenated to get word vector. It sends to Bi-LSTM getting context representation. The Multi-Head Attention mechanism extracts relevant information from different dimensions. In the classification layer, we use the softmax classifier to output the probability of each category.

The paper is organized as follows: Section 2 gives the related work. Section 3 presents the question classification method in detail. Section 4 shows the experiments and results. Section 5 summarizes the full text and give some future works.

2 Related Work

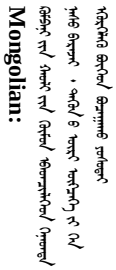
Question classification is a kind of short text classification (Alsmadi and Gan, 2019). There have been many studies on questions classification. Chinese and English, which are rich in resources, have achieved good results. The traditional method was based feature engineering such as bag of words (BOW) and n-gram. Both were combined with term frequency-inverse document frequency (TF-IDF) and other element features as text features. However, these methods ignore the context semantic information. There were some methods based machine learning, including Nearest Neighbors (NN) (Yang and Liu, 1999), Naive Bayes (McCallum et al., 1998), and Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2010). In (Wang et al., 2013), the authors utilized the external knowledge base for text classification. In recent years, researchers have tried to extract semantic information from sentences via deep learning. The combination of TextCNN (Kim, 2014), TextRNN (Liu et al., 2016), LSTM (Xiao et al., 2018), TextGCN (Yao et al., 2019), with word embedding has been widely used in text classification.

There are some researches on rare resource languages to classify questions. For example, Uyghur is also a few resource language and have complex word-formation. Parhat et al. (2019) proposed a method of Uyghur short text classification based reliable sub-word morphology. Mongolian language processing has been further developed, such as morphological segmentation (Wang et al., 2019b), spelling correction (Lu et al., 2019), named entity recognition (Wang et al., 2019a). The Mongolian question classification needs to be solved urgently.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1027-1035, Hainan, China, October 31 - Novermber 1, 2020.

Mongolian alphabet	Latin letter	Mongolian alphabet	Latin letter	Mongolian alphabet	Latin letter	Mongolian alphabet	Latin letter	Mongolian alphabet	Latin letter
ᠠ	a	ᠡ	E	ᠺ	k	ᠮ	m	ᠲ	t
ᠡᠢ	e	ᠨ	n	ᠻ	K	ᠯ	l	ᠳ	d
ᠢ	i	ᠩ	N	ᠴ	C	ᠯᠠ	L	ᠶ	y
ᠤ	q	ᠪ	b	ᠵ	Z	ᠷ	Z	ᠵ	c
ᠥ	v	ᠫ	p	ᠹ	H	ᠸ	Q	ᠵᠢ	j
ᠦ	o	ᠣ	w	ᠬ	R	ᠰ	s	ᠷᠢ	r
ᠦᠢ	u	ᠹ	f	ᠭ	g	ᠱ	x	ᠭᠠ	h

Figure 2: Comparison between Latin alphabet and Mongolian alphabet.



Latin:
kqmpani-y'in havli-y'in homun ebedcileged genedte nasv barajai , tegun-u" ori ogcege-y'i hen egurgelehu boged bvcagahv yqsqtai
Means:
Who should bear and return the debts of the company due to the sudden death of the legal person?
Category:
Company Law

Figure 3: Example of traditional Mongolian script, Latin transliterature, category tag and their meanings.

3 Model Architecture

In this section, we will introduce this model from bottom to up. The Mulit-Head Attention mechanism can fully capture the long-distance text features. But it is difficult to deal with the sequence information. The recurrent neural network can effectively obtain the context order information of sequences. It can effectively supplement the Mulit-Head Attention mechanism. As depicted in Figure 1, MA-B model is proposed by combining Bi-LSTM network with Mulit-Head Attention mechanism.

3.1 Morpheme Vector

Mongolian is a kind of agglutinative language, which consists of roots, stems and suffixes. The Chinese words need to be segmented, which is called Chinese word segmentation (Zhou et al., 2019). There are natural spaces between words in Mongolian, but morphological segmentation is needed in Mongolian because the root and stem suffixes of Mongolian words are connected with many different endings. The Mongolian word formation features result in unlimited vocabulary. This paper uses Latin to deal with Mongolian. The contrast between Latin characters and Mongolian letters is shown in Figure 2.

In this paper, we introduce Mongolian morphemes representation. The suffix is segmented by identifying a narrow uninterrupted space (NNBS) (U+202F, Latin: ”-”) to make it an independent training unit. As shown in Figure 3, after segmentation the suffix, the sentence will be turned into *“kqmpani -y’in havli -y’in homun ebedcileged genedte nasv barajai , tegun -u” ori ogcege -y’i hen egurgelehu boged bvcagahv yqsqtai”*. The length of this sentence is changed to 19 units.

The Word2vec is a common tool for training word vectors. The Word2vec (Mikolov et al., 2013) contains CBOW (Continuous Bag of Word) and Skip-gram. This paper uses the Skip-gram model to train morpheme vectors. Given a sequence of morphemes $\mathbf{m} = m_1,...,m_T \in M$. The output of the model is a probability distribution. The morpheme skip-gram model predict contextual morphemes when given current morpheme.The formula is as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(m_{t+j} \mid m_t) \tag{1}$$

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1027-1035, Hainan, China, October 31 - Novermber 1, 2020.

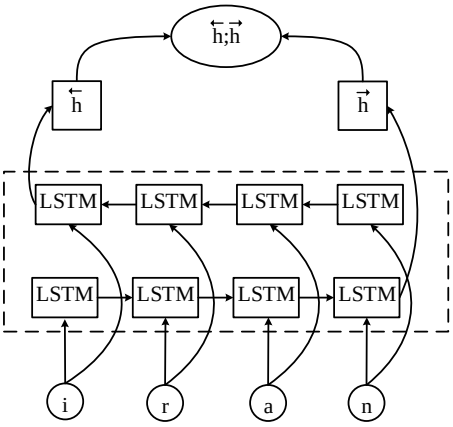


Figure 4: The character embedding of Mongolian morpheme.

where c is the size of the context window for the current central morpheme m_t. The simplest formulation of the probability $p(m_{t+j}|m_t)$ is:

$$p(o \mid c) = \frac{\exp({u_o}^T v_c)}{\sum_{m=1}^{M} \exp({u_m}^T v_c)} p \tag{2}$$

where *o* is the ids of the output morpheme, *c* is the ids of the central morpheme, *u* is the output morpheme vector, *v* is the input morpheme vector, and *M* is the morphemes set.

3.2 Character Vector

To better represent the semantic information in sentences, we use the Bi-LSTM model to learn the character embedding from training data. The character Bi-LSTM network consists of forward LSTM layer and backward LSTM layer. The forward layer can learn word prefix information. And the backward layer learns the morphological information. Both layers are connected to the same output layer. We get the character representation. As shown in Figure 4 is the structure of Bi-LSTM character embedding network.

3.3 Bi-LSTM Layer

LSTM (Hochreiter and Schmidhuber, 1997) network is a special type of recursive neural network, which can capture the context order information of the sequence and solve the problem of long dependency. LSTM is a variant of RNN. It introduces some gates to solve the gradient problem. LSTM calculates an output vector according to the current input and the output of the previous unit. The output vector is then used as input to the next unit.

LSTM is mainly composed of four parts: storage unit c_t, input gate i_t, output gate o_t, and forget gate f_t. Those gates control the proportion of history to omit or to store in the next time stamp. LSTM calculates the output vector based on the current input and the output of the previous unit, which is then used as the input of the next unit. The calculation formula is as follows:

$$\begin{aligned}
f_t &= \sigma(W_{(f)} x_t + U_{(f)} h_{t-1} + b_{(f)}) \\
i_t &= \sigma(W_{(i)} x_t + U_{(i)} h_{_{t-1}} + b_{(i)}) \\
o_t &= \sigma(W_{(o)} x_t + U_{(o)} h_{_{t-1}} + b_{(o)}) \\
c_t &= \tilde{c} + i_t \odot \tanh(W_{(c)} x_t + U_{(c)} h_{_{t-1}} + b_{(c)}) \\
\tilde{c} &= f_t \odot c_{t-1} \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{3}$$

where i_t is the input gate and o_t is the output gate. The forget gate f_t is a reset memory unit. x_t the input vector. h_t represents the hidden unit vector. σ is the point product sigmoid function. $\odot$ represents the

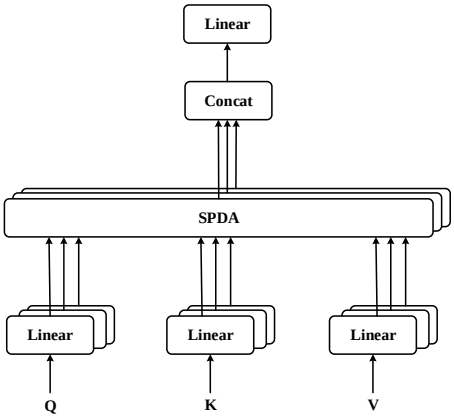


Figure 5: The flowchart of scaled dot-product attention.

corresponding multiplication of elements. W_i, W_f, W_o is the weight matrix of the input gate, the forget gate, and the output gate respectively. U_f, U_i, U_c, U_o denote the different weight matrices for hidden h_t. And b_i, b_f, b_c, b_o represent the bias.

The LSTM can only encode historical information, but it is often not enough. The paper adopted the Bidirectional LSTM network which is composed of forward LSTM and backward LSTM. So, *h* is the concatenate of $\overleftarrow{h_t}$, $\overrightarrow{h_t}$ and *h* is shown as below.

$$h = \overleftarrow{h_t} + \overrightarrow{h_t} \tag{4}$$

where $\overrightarrow{h_t}$ is the forward output vector and $\overleftarrow{h_t}$ is backward output vector.

3.4 Mulit-Head Attention Layer

In recent years, *Transformer* (Vaswani et al., 2017) model is very popular, which used in NLP tasks. It uses the Mulit-Head Attention mechanism. The Mulit-Head Attention is the optimization of the traditional attention mechanism and it is used to fully capture the features of long distance and obtain the global information. It firstly projects the input into multiple feature spaces, then compute correlation score and utilize the scores to weight context representation, finally concatenates vectors weighted as output.

The input of Mulit-Head Attention mechanism consists of *Q*(queries), *K*(keys) and *D*(dimension). The merging vector output from Bi-LSTM layer is the input of *Q, K* and *V*. Then *Q, K, V* are linearly transformed and finally input into scaled dot-product attention(SDPA). This process calculates one head at a time. As shown in Figure 5, the model independently compute dot product attention for each part $head_i$. The details are described below.

$$SDPA(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where *softmax* is a normalization function.The calculation formula is as follows:

$$softmax(g\,(Q, K)) = \frac{e^{g(Q,K)}}{\sum_i e^{g(Q,K_j)}} \tag{6}$$

where *g(Q,K)* represents the similarity between *Q* and *K*. Similarity calculation is obtained by *Q* and *K* point product operation.

Then, all the scaled dot-product attention results of *m* times, are concatenated and the value obtained by a linear transformation is used as the result of the Mulit-Head Attention model.

$$head_i = SDPA\left(QW_i^Q, QW_i^K, VW_i^V\right) \tag{7}$$

$$MA(Q, K, V) = Concat(head_1, \cdots, head_h) \tag{8}$$

where W_i^Q, W_i^K and W_i^V are projection matrices corresponding to *Q, K* and *V* respectively.

3.5 Classification Layer

Questions classification is a multi classification problem. The classification layer consists of two parts: a linear layer and a softmax layer. The text vector *h* can be used as features for questions classification.

$$y = softmax(W_h h + b_h) \tag{9}$$

We use the negative log likelihood of the correct classification as training loss.

$$L = -\sum_t \log y_{ti} \tag{10}$$

where *i* is the label of the text *t*.

4 Experiments

Our model is trained on the selected data set. By evaluating the classification results and comparing with baseline, we can evaluate the questions classification performance of the model.

4.1 Setting Up

The training data mainly comes from China Mongolian News Network, People's Daily Online (Mongolian version), China Mongolian Broadcasting Network, China Judgements Online (Mongolian version) and other web sites. After removing duplicate data and cleaning invalid data, 115688 sentences were obtained by manual correction and annotation. The data of question classification is divided into eleven categories, as shown in Table 1. We divided the dataset into train, dev and test with the percent 80%, 10% and 10%,respectively.

Label	Categories	Number	Label	Categories	Number
0	Marriage and Family	10359	6	Property Disputes	9435
1	Labor Disputes	9621	7	Infringement	11258
2	Traffic Accident	11421	8	Company Law	9900
3	Credit and Debt	9401	9	Medical Disputes	8743
4	Criminal Defense	13020	10	Administrative Litigation	13872
5	Contract Disputes	8658			

Table 1: The data is divided into eleven categories.

4.2 Evaluation Metrics

Question classification is a multi classification task, so we use *precision, recall* and F_1 as the evaluation index. These metrics are calculated as:

$$\begin{aligned} P &= \frac{TP}{TP + FP} \\ R &= \frac{TP}{TP + FN} \\ F_1 &= \frac{2PR}{P + R} \end{aligned} \tag{11}$$

where *TP* is the number of correctly predicted question sentences. *FP* is the number of sentences that predicted as question sentences, but in actuality those are negative class. If the prediction is failed, and the positive class is predicted as a false negative(*FN*). F_1 is the harmonic mean of precision and recall.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 1027-1035, Hainan, China, October 31 - Novermber 1, 2020.

4.3 Results

In this paper, TextCNN, Bi-LSTM and Attention-BiLSTM model are used as baselines. TextCNN (Kim, 2014) applies Convolutional Neural Networks(CNN) to text classification tasks. The key information in sentences is extracted by using multiple different size kernels. So it can better capture the local correlation. TextCNN is a commonly used baseline. Bi-LSTM and Attention-BiLSTM are commonly used models to extract text features. Attention is essentially an automatic weighted summation mechanism that makes the model more capable of handling long sequences.

The experiment is divided into two forms: 1) whether the combination of character vector and morpheme vector affects the performance of the model. 2) whether the introduction of Mulit-Head Attention mechanism into the model affects the performance of the model. The experimental results are shown in Table 2.

Model	Character embedding	Morpheme embedding	P(%)	R(%)	F_1(%)
TextCNN	Yes	No	82.57	79.36	80.93
TextCNN	Yes	Yes	83.27	81.42	82.33
Bi-LSTM	Yes	No	83.22	83.95	83.58
Bi-LSTM	Yes	Yes	84.56	83.93	84.24
Att-BiLSTM	Yes	No	84.67	83.91	84.31
Att-BiLSTM	Yes	Yes	85.13	84.89	85.01
MA-B	Yes	No	86.58	86.01	86.29
MA-B	Yes	Yes	**86.71**	**86.51**	**86.61**

Table 2: Comparison of experimental results.

We compare the results from the table:

1) Introducing morpheme features in the embedding layer can improve performance. The F_1 value of MA-B model remains the highest among all models. About 1.6% improvement compared with the highest Att-BiLSTM model in the baseline model.

2) In the whole model, the introduction of Mulit-Head Attention mechanism can effectively improve the model classification performance. Compared with Bi-LSTM model, our model is improved by about 2.2%. Compared with Att-BiLSTM model, our model's classification ability is also significantly enhanced.

The reasons for the above results are as follows:

1) When judging the questions categories of sentences, we mainly consider the semantic information of sentences. In Mongolian word formation, morpheme vector can learn more syntactic and semantic information. Therefore, the introduction of morpheme features into the model will have a good performance.

2) Compared with the baseline model, the advantage of MA-B model is to use BiLSTM network to obtain the internal relationship between the front and back directions of sentences and get local information. The long-distance feature is fully captured by Mulit-Head Attention mechanism, and relevant information is learned from different dimensions and representation subspaces.

5 Conclusion

In this paper, Bi-LSTM and Mulit-Head Attention mechanism are used to model Mongolian corpus texts. By combining the ability of multi head attention to obtain global information with the ability of Bi-LSTM to obtain local sequence information, a better effect has been achieved. At the same time, in order to make the model better learn the text semantic information, Mongolian morphemes representation are further introduced.

However, there is a lot of room for improvement in the field of Mongolian questions classification. From the experiment, it can be seen that the introduction of pre training morphemes features has a good effect. In the future, feature engineering can be further reduced by using pretraining language models.

At the same time, the research of Mongolian question intention recognition provides a good foundation for Mongolian question answering system in the future.

Acknowledgements

The project (Nos. 2018YFE0122900, CGZH2018125, 2019GG372, 2020GG0046) are supported by Inner Mongolia Science & Technology Plan; National Natural Science Foundation of China (Nos. 61773224); Natural Science Foundation of Inner Mongolia (Nos. 2018MS06006, 2020BS06001). Weihua Wang is the corresponding author.

References

Issa M. Alsmadi and Keng Hoon Gan. 2019. Review of short-text classification. *Int. J. Web Inf. Syst.*, 15(2):155–182.

Nello Cristianini and John Shawe-Taylor. 2010. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Xiaodong He and David Golub. 2016. Character-level question answering with attention. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1598–1607. The Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2873–2879. IJCAI/AAAI Press.

Min Lu, Feilong Bao, Guanglai Gao, Weihua Wang, and Hui Zhang. 2019. An automatic spelling correction method for classical mongolian. In *International Conference on Knowledge Science, Engineering and Management*, pages 201–214. Springer.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Dan I. Moldovan, Marius Pasca, Sanda M. Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, 21(2):133–154.

Sardar Parhat, Mijit Ablimit, and Askar Hamdulla. 2019. Uyghur short-text classification based on reliable subword morphology. *IJRIS*, 11(3):250–255.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389. The Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiang Wang, Ruhua Chen, Yan Jia, and Bin Zhou. 2013. Short text classification using wikipedia concept based document representation. In *2013 International Conference on Information Technology and Applications*, pages 471–474. IEEE.

Weihua Wang, Feilong Bao, and Guanglai Gao. 2019a. Learning morpheme representation for mongolian named entity recognition. *Neural Processing Letters*, 50(3):2647–2664.

Weihua Wang, Rashel Fam, Feilong Bao, Yves Lepage, and Guanglai Gao. 2019b. Neural morphological segmentation model for mongolian. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Lizhong Xiao, Guangzhong Wang, and Yang Zuo. 2018. Research on patent text classification based on word2vec and lstm. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 71–74. IEEE.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Jianing Zhou, Jingkang Wang, and Gongshen Liu. 2019. Multiple character embeddings for chinese word segmentation. In Fernando Emilio Alva-Manchego, Eunsol Choi, and Daniel Khashabi, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 210–216. Association for Computational Linguistics.