DIFM:An effective deep interaction and fusion model for sentence matching

Kexin Jiang
Department of Computer Science and Technology, Yanbian University.
2020010075@ybu.edu.cn

Yahui Zhao *
Department of Computer Science and Technology, Yanbian University.
yhzhao@ybu.edu.cn

Rongyi Cui
Department of Computer Science and Technology, Yanbian University.
cuirongyi@ybu.edu.cn

Abstract

Natural language sentence matching is the task of comparing two sentences and identifying the relationship between them. It has a wide range of applications in natural language processing tasks such as reading comprehension, question and answer systems. The main approach is to compute the interaction between text representations and sentence pairs through an attention mechanism, which can extract the semantic information between sentence pairs well. However, this kind of methods fail to capture deep semantic information and effectively fuse the semantic information of the sentence. To solve this problem, we propose a sentence matching method based on deep interaction and fusion. We first use pre-trained word vectors Glove and character-level word vectors to obtain word embedding representations of the two sentences. In the encoding layer, we use bidirectional LSTM to encode the sentence pairs. In the interaction layer, we initially fuse the information of the sentence pairs to obtain low-level semantic information; at the same time, we use the bi-directional attention in the machine reading comprehension model and self-attention to obtain the high-level semantic information. We use a heuristic fusion function to fuse the low-level semantic information and the high-level semantic information to obtain the final semantic information, and finally we use the convolutional neural network to predict the answer. We evaluate our model on two tasks: text implication recognition and paraphrase recognition. We conducted experiments on the SNLI datasets for the recognizing textual entailment task, the Quora dataset for the paraphrase recognition task. The experimental results show that the proposed algorithm can effectively fuse different semantic information that verify the effectiveness of the algorithm on sentence matching tasks.

1 Introduction

Natural language sentence matching is the task of comparing two sentences and identifying the relationship between them. It is a fundamental technique for a variety of tasks. For example, in the paraphrase recognition task, it is used to determine whether two sentences are paraphrased. In the text implication recognition task, it is possible to determine whether a hypothetical sentence can be inferred from a predicate sentence.

Recognizing Textual Entailment (RTE), proposed by Dagan(Dagan and Glickman, 2004), is a study of the relationship between premises and assumptions. It mainly includes entailment, contradiction, and neutrality. The main methods for recognizing textual entailment include the following: similarity-based methods(Ren et al., 2015), rule-based methods(Hu et al., 2020), alignment feature-based machine learning methods(Sultan et al., 2015), etc. However, These methods can't perform well in recognition because they didn't extract the semantic information of the sentences well. In recent years, deep learning-based methods have been effective in semantic modeling, achieving good results in many tasks in NLP(Jin et al., 2021)(Li et al., 2021)(Yang et al., 2020). Therefore, on the task of recognizing textual entailment, deep learning-based methods have outperformed earlier approaches and become the dominant recognizing textual entailment method. For example,Bowman *et al*. used recurrent neural networks to model

*Corresponding author.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

premises and hypotheses, which have the advantage of making full use of syntactic information(Bowman et al., 2015a).After that, he first applied LSTM sentence models to the RTE domain by encoding premises and hypotheses through LSTM to obtain sentence vectors(Bowman et al., 2015b). WANG *et al*.proposed mLSTM model on this basis, which focuses on splicing attention weights in the hidden states of the LSTM, focusing on the part of the semantic match between the premise and the hypothesis.The experimental results showed that the method achieved good results on the SNLI dataset(Wang and Jiang, 2016).

Paraphrase recognition is also called paraphrase detection.The task of paraphrase recognition is to determine whether two texts hold the same meaning.If they have the same meaning,they are called paraphrase pairs.Traditional paraphrase recognition methods focus on text features.However,there are problems such as low accuracy rate.Therefore,deep learning-based paraphrase recognition methods have become a hot research topic.Deep learning-based paraphrase recognition methods are mainly divided into two types; 1) calculated word vectors by neural networks,and then calculated word vector distances to determine whether they were paraphrase pairs.For example, Huang *et al*.used an improved EMD method to calculate the semantic distance between vectors and obtain the interpretation relationship(Dong-hong, 2017). 2) Directly determining whether a text pair is a paraphrased pair by a neural network model,which is essentially a binary classification algorithm.Wang *et al*.proposed the BIMPM model, which first encodes sentence pairs by a bidirectional LSTM and then matches the encoding results from multiple perspectives in both directions(Wang et al., 2017).Chen *et al*.proposed an ESIM model that uses a two-layer bidirectional LSTM and a self-attention mechanism for encoding, then it extracts features through the average pooling layer and the maximum pooling layer, and finally performs classification(Chen et al., 2017).

These models mentioned above have achieved good results on specific tasks, but most of these models have difficulty extracting deep semantic information and effectively fusing the extracted semantic information,in this paper, we propose a sentence matching model based on deep interaction and fusion .We use the bi-directional attention and self-attention to obtain the high-level semantic information.Then,we use a heuristic fusion function to fuse the low-level semantic information and the high-level semantic information to obtain the final semantic information.We conducted experiments on the SNLI datasets for the recognizing textual entailment task, the Quora dataset for the paraphrase recognition task.,The results showed that the accuracy of the proposed algorithm on the SNLI test set is 87.1%, and the accuracy of the Quora test set is 86.8%. Our contributions can be summarized as follows:

- We propose a sentence matching model based on deep interaction and fusion. It introduces bidirectional attention mechanism into sentence matching task for the first time.
- We propose a heuristic fusion function. It can learn the weights of fusion by neural network to achieve deep fusion.
- We evaluate our model on two different tasks and Validate the effectiveness of the model.

2 BIDAF model based on bi-directional attention flow

In the task of extractive machine reading comprehension, Seo *et al*.first proposed a bi-directional attention flow model BIDAF (Bi-Directional Attention Flow) for question-to-article and article-to-question(Seo et al., 2016). Its structure is shown in Figure 1.

The model mainly consists of an embed layer, a contextual encoder layer, an attention flow layer, a modeling layer, and an output layer. After the character-level word embedding and the pre-trained word vector Glove word embedding, the contextual representations X and Y of the article and the question are obtained by a bidirectional LSTM, respectively.The bi-directional attention flow between them is computed, and it proceeds as follows:

a) The similarity matrix between the question and the article is calculated.The calculation formula is shown in Eq.1.

$$K_{tj} = W^T [X_{:t}; Y_{:j}; X_{:t} \odot Y_{:j}] \tag{1}$$

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

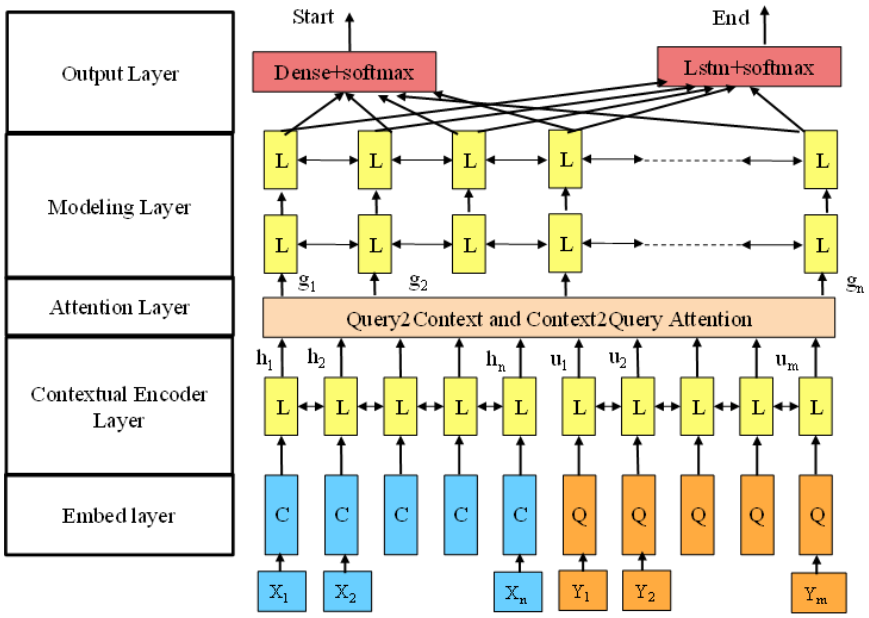

Figure 1: Bi-Directional Attention Flow Model

where K_{tj} is the similarity of the t-th article word to the j-th question word, $X_{:t}$ is the t-th column vector of X, $Y_{:j}$ is the j-th column vector of Y, and W is a trainable weight vector.

b) Calculating the article-to-question attention. Firstly, the normalization operation is performed on the above similarity matrix, and then the weighted sum of the problem vector is calculated to obtain the article-to-problem attention, which is calculated as shown in Eq.2.

$$\begin{aligned} x_t &= soft\max(K) \\ \hat{Y}_{:t} &= \sum_j x_{tj} Y_{:j} \end{aligned} \tag{2}$$

c) Query-to-context (Q2C) attention signifies which context words have the closest similarity to one of the query words and are hence critical for answering the query. We obtain the attention weights on the context words by $y = softmax(max_{col}(K)) \in R^T$, where the maximum function $\max_{col}$ is performed across the column. Then the attended context vector is $\hat{x} = \sum_t y_t X_{:t}$.This vector indicates the weighted sum of the most important words in the context with respect to the query. $\hat{x}$ is tiled T times across the column, thus giving $\hat{X} \in R^{2d*T}$.

d) Fusion of bidirectional attention streams. The bidirectional attention streams obtained above are stitched together to obtain the new representation, which is calculated as shown in Eq.3.

$$L_{:t} = \left[X_{:t}; \hat{Y}_{:t}; X_{:t} \odot \hat{Y}_{:t}; X_{:t} \odot \hat{X}_{:t}\right] \tag{3}$$

We builds on this work by looking at sentence pairs in a natural language sentence matching task as articles and problems for reading comprehension.We use the bi-directional attention and self-attention to obtain the high-level semantic information.Then,we use a heuristic fusion function to fuse the low-level semantic information and the high-level semantic information to obtain the final semantic information.

3 Method

In this section, we describe our model in detail. As shown in Figure 2, our model mainly consists of an embedding layer, a contextual encoder layer, an interaction layer, a fusion layer, and an output layer.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

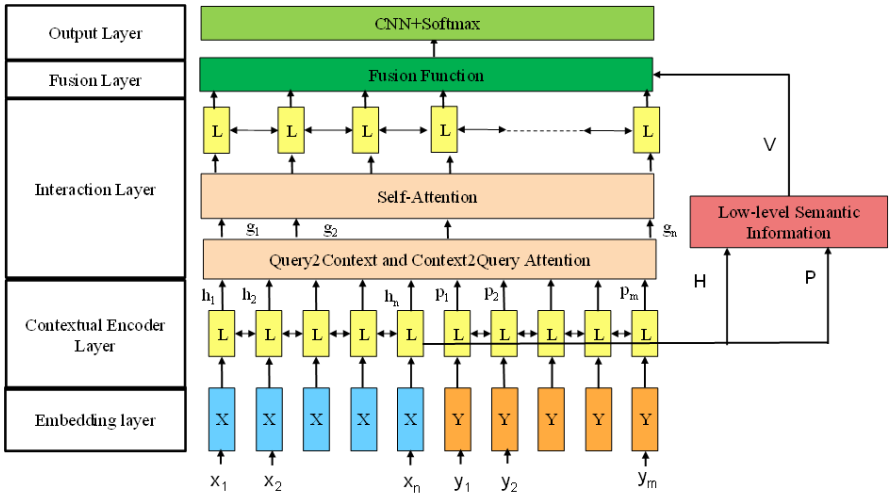

Figure 2: Overview of the architecture of our proposed DIFM model.It consists of an embedding layer, a contextual encoder layer, an interaction layer, a fusion layer, and an output layer.

3.1 Embedding Layer

The purpose of the embedding layer is to map the input sentence A and sentence B into word vectors.The traditional mapping method is one-hot encoding.However,it is spatially expensive and inefficient, so we use pre-trained word vectors for word embedding.These word vectors are constant during training.

Since the text contains unregistered words, we also use character-level word vector embedding.Each word can be seen as a concatenation of characters and characters, and then we use LSTM to get character-level word vectors. It can effectively handle unregistered words.

We assume that the pre-trained word vector for word h is h_w,and character-level word vector is h_c,we splice the two vectors and use a two-tier highway network(Zilly et al., 2017) to get the word vector representation of word h:$h = [h_1; h_2] \in R^{d_1+d_2}$,where d_1 is the dimension of Glove word embedding and d_2 is the dimension of character-level word embedding.Finally,we obtain the word embedding matrix $X \in R^{n*(d_1+d_2)}$ for sentence A and the word embedding matrix $Y \in R^{m*(d_1+d_2)}$ for sentence B, where n,m represent the number of words in sentence A and sentence B.

3.2 Contextual Encoder Layer

The purpose of the contextual encoder layer is to fully exploit the contextual relationship features of the sentences. We use bidirectional LSTM for encoding which can mine the contextual relationship features of the sentences.Then,we can obtain its representation $H \in R^{2d*n}$ and $P \in R^{2d*m}$, where d is the hidden layer dimension.

3.3 Interaction Layer

The purpose of the interaction layer is to extract the effective features between sentences.In this module,we can obtain low-level semantic information and high-level semantic information.

3.3.1 low-level semantic information

The purpose of this module initially fuses two sentences to get the low-level semantic information.We first calculate the similarity matrix S of the context-encoded information H and P,which is shown in Eq.4.

$$S_{ij} = {W_s}^T[h; p; h \odot p] \tag{4}$$

where S_{ij} denotes the similarity between the i-th word of H and the j-th word of P, W_s is weight matrices, h is the i-th column of H , and p is the j-th column of P. Then, we calculate the low-level semantic information V of A and B,which is shown in Eq.5.

$$V = P \cdot softmax(S^T) \tag{5}$$

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

3.3.2 high-level semantic information

The purpose of this module is mine the deep semantics of the text, and to generate high-level semantic information.In this module,we frist calculate the bidirectional attention of H and P that is the attention of $H \rightarrow P$ and $P \rightarrow H$. It is calculated as follows.

$H \rightarrow P$: The attention describes which words in the sentence P are most relevant to H.The calculation process is as follows; firstly, each row of the similarity matrix is normalized to get the attention weight, and then the new text representation $Q \in R^{2d*n}$ is obtained by weighted summation with each column of P,which is calculated as shown in Eq.6.

$$\begin{aligned} \alpha_t &= softmax(S_{t:}) \in R^m \\ q_{:t} &= \sum_j \alpha_{tj} P_{:j} \end{aligned} \tag{6}$$

where $q_{:t}$ is the t-th column of Q.

$P \rightarrow H$: The attention indicates which words in H are most similar to P.The calculation process is as follows: firstly,the column with the largest value in the similarity matrix $\boldsymbol{S}$ is taken to obtain the attention weight, then the weighted sum of H is expanded by n time steps to obtain $C \in R^{2d*n}$,which is calculated as shown in Eq.7.

$$\begin{aligned} b &= softmax(\max_{col}(S)) \in R^n \\ c &= \sum_t b_t H_{t:} \in R^{2d} \end{aligned} \tag{7}$$

After obtaining the attention matrix Q of $H \rightarrow P$ and the attention matrix C of $P \rightarrow H$,we splice the attention in these two directions by a multilayer perceptron.Finally,we get the spliced contextual representation G, which is calculated as shown in Eq.8.

$$\begin{aligned} G_{:t} &= \beta(C_{:t}, H_{:t}, Q_{:t}) \\ \beta(c, h, q) &= [h; q; h \odot q; h \odot c] \in R^{8d} \end{aligned} \tag{8}$$

Then,we calculate its self-attention(Vaswani et al., 2017) ,which is calculated as shown in Eq.9.

$$\begin{aligned} E &= G^T G \\ Z &= G \cdot softmax(E) \end{aligned} \tag{9}$$

Finally, we pass the above semantic information Z through a bi-directional LSTM to obtain high-level semantic information U.

3.4 Fusion Layer

The purpose of the fusion layer is to fuse the low-level semantic information V and the high-level semantic information U. We innovatively propose a heuristic fusion function, it can learn the weights of fusion by neural network to achieve deep fusion.We fuse V and U to obtain the text representation $L = fusion(U, V) \in R^{n*2d}$, where the fusion function is defined as shown in Eq.10:

$$\begin{aligned} \widetilde{x} &= \tanh(W_1[x; y; x \odot y; x - y]) \\ g &= sigmoid(W_2[x; y; x \odot y; x - y]) \\ z &= g \odot \widetilde{x} + (1 - g) \odot x \end{aligned} \tag{10}$$

Where W_1 and W_2 are weight matrices, and g is a gating mechanism to control the weight of the intermediate vectors in the output vector. In this paper, x refers to U and y refers to V.

3.5 Output Layer

The purpose of the output layer is to output the results.In this paper, we use a linear layer to get the results of sentence matching. The process is shown in Eq.11.

$$y = softmax(\tanh(ZW + b)) \tag{11}$$

where both W and b are trainable parameters. Z is the vector after splicing its first and last vectors.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

4 Experimental results and analysis

In this section,we validate our model on two datasets from two tasks.We first present some details of the model implementation, and secondly, we show the experimental results on the dataset. Finally, we analyze the experimental results.

4.1 Experimental details

4.1.1 Loss function

In this paper, the cross-entropy loss function can be chosen as shown in Eq.12.

$$loss = -\sum_{i=1}^{N}\sum_{k=1}^{K} y^{(i,k)} \log \hat{y}^{(i,k)} \tag{12}$$

where N is the number of samples, K is the total number of categories and $\hat{y}^{(i,k)}$ is the true label of the i-th sample.

4.1.2 Dataset

In this paper, we use the natural language inference datasets SNLI, and the paraphrase recognition dataset Quora to validate our model. The SNLI dataset contains 570K manually labeled and categorically balanced sentence pairs. The Quora question pair dataset contains over 400k pairs of data that each with binary annotations, with 1 being a duplicate and 0 being a non-duplicate.The statistical descriptions of SNLI and Quora data are shown in Table 1.

Table 1: The statistical descriptions of SNLI and Quora

dataset	train	validation	test
SNLI	550152	10000	10000
Quora	384290	10000	10000

Table 2: Values of Hyper Parameters

Hyper Parameters	Values
Glove dimension	300
Character embedding dimension	100
Hidden dimension	200
learning rate	0.0005
Optimizer	Adam
Dropout	0.2
activation function	ReLU
Epoch	30
Batch size	128

4.1.3 parameter settings

This experiment is conducted in a hardware environment with a graphics card RTX5000 and 16G of video memory.The system is Ubuntu 20.04, the development language is Python 3.7, and the deep learning framework is Pytorch 1.8.

In the model training process, a 300-dimensional Glove word vector are used for word embedding, and the maximum length of text sentences is set to 300 and 50 words on the SNLI and Quora datasets, respectively. The specific hyperparameter settings are shown in Table 2.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

4.2 Experimental results and analysis

We compare the experimental results of the sentence matching model based on deep interaction and fusion on the SNLI dataset with other published models.The evaluation metric we use is the accuracy rate.The results are shown in Table 3. As can be seen from Table 3, our model achieves an accuracy rate of 0.871 on the SNLI dataset, which achieves better results in the listed models.Compared with the LSTM, it is improved by 0.065. Compared with Star-Transformer model,it is improved by 0.004. Compared with some other models, it is observed that our model is better than the others model.

Table 3: The accuracy(%) of the model on the SNLI test set.Results marked with [a] are reported by Bowman et al.(Bowman et al., 2016),[b] are reported by Han et al.(Han et al., 2019), [c] are reported by Shen et al.(Shen et al., 2018), [d] are reported by Borges et al.(Borges et al., 2019), [e] are reported by Guo et al.(Guo et al., 2019), [f] are reported by Mu et al.(Mu et al., 2018).

Model	Acc
300D LSTM encoders[a]	80.6
DELTA[b]	80.7
SWEM-max[c]	83.8
Stacked Bi-LSTMs[d]	84.8
Bi-LSTM sentence encoder[d]	84.5
Star-Transformer[e]	86.0
CBS-1+ESIM[f]	86.7
DIFM	**87.1**

We conduct experiments on the Quora dataset, and the evaluation metric is accuracy. The experimental results on the Quora dataset are shown in Table 4. As can be seen from Table 4,the accuracy of our method on the test set is 0.868.The experimental results improve the accuracy by 0.054 compared to the traditional LSTM model.Compared with the enhanced sequential inference model ESIM,it is improved by 0.004. The experimental results achieved good results compared to some current popular deep learning methods.Our model achieve relatively good results in both tasks, which illustrates the effectiveness of our model.

Table 4: The accuracy(%) of the model on the Quora test set.Results marked with [g] are reported by Yang et al.(Yang et al., 2021), [h] are reported by He et al.(He and Lin, 2016) , [i] are reported by Zhao et al.(Zhao et al., 2021), [j] are reported by Chen et al.(Chen et al., 2017).

Model	Acc
LSTM	81.4
RCNN[g]	83.6
PWIM[h]	83.4
Capsule-BiGRU[i]	86.1
ESIM[j]	85.4
DIFM	**86.8**

4.3 Ablation experiments

To explore the role played by each module, we conduct an ablation experiment on the SNLI dataset .Without using the fusion function, which means that the low-level semantic information are directly spliced with the high-level semantic information. The experimental results are shown in Table 5.

We first verify the effectiveness of character embedding. Specifically,we remove the character embedding for the experiment, and its accuracy drops by 1.5 percentage points, proving that character embedding plays an important role in improving the performance of the model.

Proceedings of the 21st China National Conference on Computational Linguistics, pages 738-747, Nanchang, China, October 14 - 16, 2022.

Table 5: Ablation study on the SNLI validation dataset

Model	Acc(%)
DIFM	87.1
w/o character embedding	85.6 (↓1.5)
w/o low-level semantic information	85.9 (↓1.2)
w/o high-level semantic information	79.5 (↓**7.6**)
w/o fusion	86.1(↓1.0)
w/o self-attention	58.8(↓1.3)
w/o $P \rightarrow H$	84.6(↓2.5)
w/o $H \rightarrow P$	86.2(↓0.9)

In addition,we verify the effectiveness of the semantic information and fusion modules. We removed low-level semantic information and high-level semantic information from the original model, and its accuracy dropped by 1.2 percentage points and 7.6 percentage points. At the same time, we remove the fusion function, and its accuracy drops by about 1.0 percentage points. It shows that the different semantic information and the fusion function are beneficial to improve the accuracy of the model, with the high-level semantic information being more significant for the model.

Finally,we verify the effectiveness of each attention on the model.We remove the attention from P to H, the attention from H to P, and the self-attention module respectively. Their accuracy rates decreased by 2.5 percentage points, 0.9 percentage points, and 1.3 percentage points. It shows that all the various attention mechanisms improve the performance of the model, with the P to H attention being more significant for the model.

The ablation experiments show that each component of our model plays an important role, especially the high-level semantic information module and the P to H attention module, which have a greater impact on the performance of the model. Meanwhile, the character embedding and fusion function also play an important role in our model.

5 Conclusion

we investigate natural language sentence matching methods and propose an effective deep interaction and fusion model for sentence matching. Our model first uses the bi-directional attention in the machine reading comprehension model and self-attention to obtain the high-level semantic information.Then,we use a heuristic fusion function to fuse the semantic information that we get.Finally, we use a linear layer to get the results of sentence matching .We conducted experiments on SNLI and Quora datasets. The experimental results show that the model proposed in this paper can achieve good results in two tasks.In this work,we find that our proposed interaction module and fusion module occupie the dominant position and have a great impact on our model.However,Our model is not as powerful as the pre-trained model in terms of feature extraction and lacks external knowledge.The next research work plan will focus on the following two points: 1) we use more powerful feature extractors, such as BERT pre-trained model as text feature extractors; 2) the introduction of external knowledge will be considered. For example, WordNet, an external knowledge base, contains many sets of synonyms, and for each input word, its synonyms are retrieved from WordNet and embedded in the word vector representation of the word to further improve the performance of the model.

Acknowledgements

This work is supported by National Natural Science Foundation of China [grant numbers 62162062]. State Language Commission of China under Grant No. YB135-76, scientific research project for building world top discipline of Foreign Languages and Literatures of Yanbian University under Grant No. 18YLPY13. Doctor Starting Grants of Yanbian University [2020-16], the school-enterprise cooperation project of Yanbian University [2020-15].

References

Luís Borges, Bruno Martins, and Pável Calado. 2019. Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *Journal of Data and Information Quality (JDIQ)*, 11(3):1–26.

Samuel Bowman, Christopher Potts, and Christopher D Manning. 2015a. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 12–21.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015b. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 632–642. Association for Computational Linguistics (ACL).

Samuel R Bowman, Raghav Gupta, Jon Gauthier, Christopher D Manning, Abhinav Rastogi, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 1466–1477. Association for Computational Linguistics (ACL).

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004:26–29.

HJPJ Dong-hong. 2017. Convolutional network-based semantic similarity model of sentences. *Journal of South China University of Technology (Natural Science)*, 45(3):68–75.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325.

Kun Han, Junwen Chen, Hui Zhang, Haiyang Xu, Yiping Peng, Yun Wang, Ning Ding, Hui Deng, Yonghu Gao, Tingwei Guo, et al. 2019. Delta: A deep learning based language technology platform. *arXiv preprint arXiv:1908.01853*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pages 937–948.

Chaowen Hu, Changxing Wu, and Yalian Yang. 2020. Extended s-lstm based textual entailment recognition. *Journal of Computer Research and Development*, 57(7):1481–1489.

Jing Jin, Yahui Zhao, and Rongyi Cui. 2021. Research on multi-granularity ensemble learning based on korean. In *The 2nd International Conference on Computing and Data Science*, pages 1–6.

Feiyu Li, Yahui Zhao, Feiyang Yang, and Rongyi Cui. 2021. Incorporating translation quality estimation into chinese-korean neural machine translation. In *China National Conference on Chinese Computational Linguistics*, pages 45–57. Springer.

Norman Mu, Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. 2018. Parameter re-initialization through cyclical batch size schedules. *arXiv preprint arXiv:1812.01216*.

Han Ren, Yaqi Sheng, and Wenhe Feng. 2015. Recognizing textualentailmentbasedonknowledgetopicmodels. *Journal of Chinese Information Processing*, 29(6):119–127.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Feature-rich two-stage logistic regression for monolingual alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 949–959.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Feiyang Yang, Yahui Zhao, and Rongyi Cui. 2020. Recognition method of important words in korean text based on reinforcement learning. In *China National Conference on Chinese Computational Linguistics*, pages 261–272. Springer.

Dezhi Yang, Xianxin Ke, and Qichao Yu. 2021. A question similarity calculation method based on rcnn. *Journal of Computer Engineering and Science*, 43(6):1076–1080.

Qi Zhao, Yanhui Du, and Tianliang Lu. 2021. Algorithm of text similarity analysis based on capsule-bigru. *Journal of Computer Engineering and Applications*, 57(15):171–177.

Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutnık, and Jürgen Schmidhuber. 2017. Recurrent highway networks. In *International conference on machine learning*, pages 4189–4198. PMLR.