Research on Recognition Textual Entailment with Integrated Attention Mechanism and Lexical Information

Dequan Zheng
School of Computer and Information Engineering
Harbin University of Commerce
Harbin,China
dqzheng@hrbcu.edu.cn

Haochen Liu
School of Computer and Information Engineering
Harbin University of Commerce
Harbin,China
lhc835535851@gmail.com

Feng Yu
School of Computer and Information Engineering
Harbin University of Commerce
Harbin,China
yufeng@hrbcu.edu.cn

***Abstract*—Recognition Textual Entailment is a fundamental research direction in natural language processing. However, due to the complexity of sentence structure and the multiple parts of speech of words, it affects the model's deep understanding of sentences. From a linguistic perspective, different parts of speech have different understandings of sentences. We believe that a complete sentence consists of two parts: sentence structure and syntactic structure. In response to this issue, this article proposes a recognition textual entailment model that integrates attention mechanism and part of speech tagging. NLTK (Natural Language Toolkit) is used to annotate the part of speech of sentences, and the part of speech information is interactively fused with the sentence features extracted by the Conformer model. After part of speech tagging, the model can better grasp the syntactic structure of sentences. Experiments have shown that the above method combines sentence structure and syntactic structure, which can effectively enhance the model's understanding of deep semantics and improve its accuracy compared to classical models.**

Keywords—recognition textual entailment, conformer, part-of-speech tagging, attention mechanism,

I. INTRODUCTION

Recognition Textural Entailment (RTE) is a task in the field of Natural Language Processing (NLP) aimed at testing the performance of text inference. Recognition Textural Entailment, also known as natural language inference, can be applied in multiple fields of natural language processing. Textural Entailment was first proposed by Dagan [1]. Given the given premise sentence P and hypothesis sentence H, the logical relationship between the two sentences can be determined, and the relationship can be divided into entailment, contradiction, and neutral. If people can infer the validity of the hypothesis sentence H from the premise sentence P through semantic understanding of the sentence, it is said that the premise sentence P implies the hypothesis sentence H, denoted as P → H. If people can infer the invalidity of the hypothesis sentence H from the premise sentence P through semantic understanding of the sentence, it is said that the premise sentence P contradicts the hypothesis sentence H. If it is not possible to determine the relationship between the two through semantic understanding of the sentence, it is said that the premise sentence P and the hypothesis sentence H are neutral. Table I shows an example of recognition textual entailment. Textual entailment technology is widely used in natural language processing tasks such as text summarization, relation extraction, Machine TranslationScoring System, and question-answering systems [2-5]. Today, with the rapid development of large models, textual entailment faces many challenges, such as improving the understanding and response abilities of large language models in dialogue.

TABLE I. TEXTUAL ENTAILMENT DATA EXAMPLES

Premise	Hypothesis	Relation
P_1: Four boys playing on a carousel in the park	H_1: People are playing in the park	entailment
P_2: Two young women sitting on the floor chatting	H_2: Two young women running together	contradiction
P3: A group of people celebrate during the festival	H3: People are celebrating the Lunar New Year	neutral

II. INTRODUCTION TO RELEVANT TECHNIQUES

In the early stages of recognition textual entailment, the main approaches were based on similarity-based recognition textual entailment [6] and text alignment-based recognition textual entailment methods. With the advancement of deep learning, researchers proposed methods involving sentence encoding and feature interaction. Sentence encoding involves encoding two input sentences, generating vector representations for each, and performing operations on these sentence vectors for comparison to determine the entailment relationship between the two sentences. Bowman [7] employed Long Short-Term Memory (LSTM) networks and achieved an accuracy of 77.6% on the English dataset SNLI.Feature interaction methods utilize attention mechanisms to align and compare semantic information in the premise and hypothesis sentences. Rocktäschel [8] introduced attention mechanisms to address recognition textual entailment tasks. Chen proposed the Enhanced Sequential Inference Model (ESIM) [9] and the Knowledge-Based Inference Model (KIM) [10] based on structural knowledge. Additionally, the Google team introduced the Transformer model [11], retaining attention mechanisms and

introducing an encoding structure that enables parallel computation, significantly reducing training time. Devlin [12] utilized a bidirectional Transformer model to encode textual representations. This model was pretrained on a deep Transformer structure using massive unsupervised data and fine-tuned on recognition textual entailment tasks.In recent years, aided by the release of large-scale datasets like SNLI [13] and MultiNLI [14], along with the development of pretrained language models, many scholars have explored the integration of external knowledge into language models, such as knowledge graphs or linguistic knowledge. Knowledge Graph is a model that represents knowledge based on graph data structures. It consists of nodes, attributes, and edges. Nodes represent each entity, attributes describe the characteristics of the entity, and edges represent the relationships between entities. Entities can represent specific people, things, concepts, words, etc. A relationship represents a certain connection between entities. The essence of knowledge graph is network, which connects entities and enables computers to understand human knowledge and improve problem-solving ability. Language knowledge, such as part of speech knowledge, synonyms, antonyms, and hierarchical relationships between words, can be incorporated into language models to better capture semantic features. In recent years, researchers have focused on deep semantic understanding and information extraction of text.

Part-of-speech tagging (POS tagging) involves annotating the grammatical category of each word in a sentence. POS tagging is a fundamental research area in natural language processing and finds applications in fields such as named entity recognition, machine translation, and syntactic analysis [15-16].

Recognition textual entailment is a complex reasoning process, and existing research methods struggle to handle issues like long sentences, syntactically complex structures, and sentences with ambiguous words. In this paper, we propose a recognition textual entailment approach that combines attention mechanisms with part-of-speech tagging. This method effectively enhances the model's ability to handle intricate sentences.

III. Model

This paper proposes a recognition textual entailment model that integrates attention mechanisms and part-of-speech tagging, consisting of four components as illustrated in Figure 1. Encoder Module: Building upon the transformer architecture introduced by Gulati [17], we incorporate a CNN network. While Transformers excel in global content interaction, CNNs effectively capture local features. The sentence is fed into the encoder to enable each word to better utilize contextual information. Part-of-Speech Tagging Encoding Module: We believe a complete sentence comprises two aspects - sentence structure (such as word order) and sentence syntax (such as part-of-speech). Nouns and verbs play vital roles in sentences, greatly influencing the comprehension of semantic information. Interaction Fusion Module: This involves the interaction and fusion of part-of-speech tagging encoded information with sentence features extracted by the conformer module. Classification Module: This module classifies the entailment relationship of sentence pairs into three categories: entailment, neutral, and contradiction.

A. Encoder Layer

In this paper, we utilize the conformer module from Gulati [17] for sentence encoding. Conformer is an enhancement of the transformer encoder that incorporates a Convolutional Neural Network (CNN). While the Transformer model excels at extracting features from long sequential text, CNNs are adept at capturing local features.

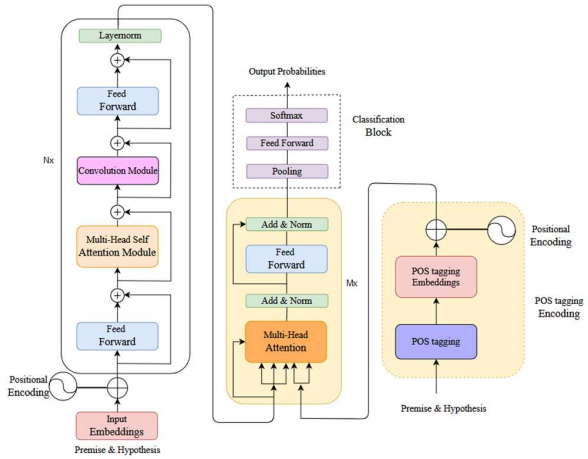

Fig.1. Textual Entailment Recognition Model with Integrated Attention Mechanism and Lexical Information

By combining the strengths of both, we can achieve a more comprehensive modeling of both local and global features in textual content.

As shown in the left part of the diagram, the input consists of a premise sentence P=$\{p_1,p_2,p_3,\ldots,p_m\}$ and a hypothesis sentence H=$\{h_1,h_2,h_3,\ldots,h_n\}$, where m and n are the numbers of words in the premise and hypothesis sentences respectively. After concatenating the premise and hypothesis sentences, tokenization yields $X=\{[CLS],\{T_1^p,T_2^p,\ldots,T_i^p,[SEP],\ T_1^h,T_2^h,\ldots,T_j^h[SEP]\}$. This is transformed into high-dimensional vectors through tokenization, resulting in $X_{TokenEmbedding}=\{[CLS],\{x_1^p,x_2^p,\ldots,x_i^p,[SEP],x_1^h,x_2^h,\ldots,x_j^h[SEP]\}$ where [SEP] is used to separate the premise and hypothesis sentences. The model employs a Position Embedding mechanism.

$$PE(pos,2i)=sin\left(\frac{POS}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE(pos,2i+1)=sin\left(\frac{POS}{10000^{\frac{2i}{d_{model}}}}\right) \quad (1)$$

As depicted in Formula (1), The input embedding of this model is divided into three types, similar to BERT, which are position embedding, token embedding, and segment embedding. Where i represents the dimension of word vectors, the positional encoding introduces temporal information to the attention. Additionally, the segmentation mechanism of the sentences is encoded. $X\in R^{L\times d}$

$$X_I=X_{PositionEmb}+X_{TokenEmb}+X_{SegmentEmb} \quad (2)$$

Building upon the foundation of multi-head self-attention, the Conformer introduces Relative Positional Embedding, which enhances the ability of the self-attention mechanism to handle varying sentence lengths and demonstrates greater robustness. Furthermore, Conformer extends from a single layer of the Transformer encoder's Feed Forward component to two consecutive layers of Feed-Forward. This modification enhances the model's representational capacity and non-linear transformation capabilities, ultimately improving model performance for processing intricate input texts. For the Conformer with input $x_i$and output y_i, the following formulas apply:

$$\tilde{x}_i = x_i + \frac{1}{2}FFN(x_i) \tag{3}$$

$$x_i' = \tilde{x}_i + MHSA(\tilde{x}_i) \tag{4}$$

$$x_i'' = x_i' + Conv(x_i') \tag{5}$$

$$y_i = Layernorm(x_i'' + \frac{1}{2}FFN(x_i'')) \tag{6}$$

Where FFN stands for Feed-Forward Network, MHSA represents Multi-Head Self-Attention mechanism, and Conv corresponds to the Convolution module.

The Feed-Forward Network consists of two linear transformations with an intermediate non-linear activation, as expressed by the following formula:

$$ReLU(x) = max(0,x) \tag{7}$$

$$Linear(x) = w_1 x + b_1 \tag{8}$$

$$FNN(x) = linear_1(ReLU(Linear_2(x)) \tag{9}$$

The formula for the Multi-Head Attention is as follows, where q, k, and v are obtained through linear transformations. H represents the number of heads in the multi-head self-attention, d_{h1} signifies the dimension of the matrix, and Transpose denotes matrix operations, resulting in the computation of the Multi-Head Attention output X_a.

$$X_{score}^a = Softmax\left(\frac{qk^T}{\sqrt{d_{h1}}}\right), X_{score}^a \in R^{H\times L\times L} \tag{10}$$

$$X_a = Transpose(X_{score}^a v), X_a \in R^{L\times d} \tag{11}$$

The formula for the convolution is presented as follows:

$$m_i = f(w_2 x_i + b_2) \tag{12}$$

B. Part-of-Speech Tagging Layer

The part-of-speech tagging module comprises two processes: (1) extracting word part-of-speech information from the sentence using NLTK tools and (2) encoding the part-of-speech information.

We employ the NLTK tools to process sentences, annotating the parts of speech of individual words, such as nouns, verbs, adjectives, prepositions, pronouns, and numerals. NLTK is a natural language processing toolset based on Python. Based on statistical models trained on data, given the context of a word, it predicts the part of speech of other words. There are many pre-trained part-of-speech taggers in the NLTK library, such as nltk.pos_. The Penn Treebank tagger in the tag() function is used to annotate the part of speech of sentences. The NLTK part of speech tagger has 36 categories.For instance, the sentence "A monkey is wading through a river" can yield the results shown in Table II through the use of NLTK tools.

Encoding part-of-speech information is similar to encoding input vectors, involving both part-of-speech features and positional features. Firstly, vector embeddings are employed, resulting in the formula:

TABLE II. PART-OF-SPEECH TAGGING

input text	pos tagging	introduction	ID
A	DT	Determiner	3
monkey	NN	Noun,singular or mass	12
is	VBZ	Verb,3rd person singular present	32
wading	VBG	Verb,gerund or present	29
through	IN	participle	6
a	DT	Preposition	3
river	NN	Determiner	12
.	SYM	Noun,singular or mass	24
		Symbol	

$$X_2 = X_{POS\ tagging\ Emb} + X_{PositionEmb}, X \in R^{L\times d} \tag{13}$$

Where both $X_{POS\ tagging\ Emb}$ and $X_{PositionEmb}$ have dimensions of d.

C. Interaction Layer

The function of the Attention Interaction Module is to synergistically integrate the sentence features extracted by the conformer and the encoding information from part-of-speech tagging into a multi-head attention mechanism. Specifically, this attention layer comprises m interaction modules, designed to comprehensively extract feature information from various dimensions. Each interaction module consists of two sub-layers: the Multi-Head Attention layer and the Feed Forward layer. Both sub-layers employ residual connections and layer normalization for stacking.

The interaction module calculates attention weights separately for the two components and then computes the sum of the attention weights for both parts. The formula is shown below:

$$X_{score}^b = Softmax\left(\frac{q^a k^{aT}}{\sqrt{d_{h2}}}\right) + Softmax\left(\frac{q^b k^{bT}}{\sqrt{d_{h3}}}\right), X_{score}^b \in R^{H_2\times L\times L} \tag{14}$$

$$X^b = Transpose(X_{score}^b v^a), X^b \in R^{L\times d} \tag{15}$$

Where q^a, k^a, and v^a are obtained through three linear transformations from the output vectors of the conformer, with dimensions $H_2\times L\times d_{h2}$, where H_2 represents the number of attention heads and d_{h2} represents the matrix dimension. q^b and k^b are obtained through two linear transformations from the part-of-speech tagging encoded vectors, with dimensions $H_2\times L\times d_{h3}$. The interaction is the normalized sum of the two part-wise attention mechanisms. X^b is the final output of the interaction module.

D. Classification Layer

Following the interaction layers, the prediction layer consists of a pooling layer, a Feed Forward layer, and a Softmax layer, responsible for making final predictions. Initially, a conformer sentence-level language model is used, with a [CLS] token added at the beginning of each sentence pair. Subsequently, deep encoding is applied to the [CLS] token. Since the conformer can capture global information disregarding spatial and distance constraints,

the output vector of [CLS] is directly utilized as the pooling result. This pooled result then passes sequentially through the Feed Forward layer and the Softmax layer. The labels encompass three possible outcomes: 0 represents entailment, 1 represents contradiction, and 2 represents neutrality. As illustrated in the formula, where y denotes the true label and ŷ signifies the model's predicted label, the objective is to minimize the discrepancy between the two.

TABLE III. EXPERIMENTAL DATASETS

Dataset	Number	
SNLI	train	549367
	dev	9842
	test	9842
MultiNLI	train	392703
	dev	20000
	test	20000

The computation process of the prediction layer is presented in the formula:

$$\hat{y} = Softmax\left(FNN\left(x_{first}\right)\right), x_{first} \in R^{d_c} \quad (16)$$

$$loss = ylog+(1-y)log(1-\hat{y}) \quad (17)$$

IV. EXPERIMENT

A. Experimental Data

This paper employs the SNLI dataset and the MultiNLI dataset. The original SNLI dataset, publicly released by Stanford University in 2015, comprises 570,152 sentence pairs. Each sentence pair is annotated with entailment, contradiction, neutrality, or "-", where "-" signifies an unspecified relationship label according to human annotators. To establish a consistent evaluation standard, data containing "-" labels were removed during data preprocessing.

The MultiNLI dataset is an extended version of the SNLI dataset, featuring 432,702 sentence pairs. Similarly, it includes entailment, contradiction, and neutrality labels, enhancing both the corpus scope and reasoning complexity. The table III presents detailed information about the experimental datasets:

The experiments in this paper use Accuracy as the evaluation metric, defined specifically as shown in the formula:

$$Accuracy = \frac{N_{corre}}{N_{predicted}} \times 100\% \quad (18)$$

Where $N_{correct}$ represents the number of correctly predicted sentence pairs with the correct relationship, and $N_{predicted}$ is the total number of predicted sentence pairs.

B Experimenttal Setup

The experiments were conducted using the Python programming language and implemented with the PyTorch deep learning framework. The laboratory equipment consisted of an i7-9700k CPU and an NVIDIA T4 GPU.

The encoder module employed the Conformer model structure, with the following parameter settings: the initial learning rate for model training was set to 2e-5, batch size was set to 32, the maximum fixed length of sentence pairs was 115, the maximum value for positional markers was 512, word vector dimension was 768, and part-of-speech tagging vector dimension was 120. To prevent overfitting, dropout layers were incorporated into the model. In the Conformer pretraining phase, as well as between hidden layers and within attention layers, the initial dropout rate was set to 0.1. In the attention interaction module, the initial dropout rate between hidden layers was set to 0.1, while no dropout was employed within the attention interaction module itself.

TABLE IV. PERFORMANCE OF DIFFERENT MODELS ON THE SNLI DATASET

Modle	Training Set (%)	Test Set (%)
300D LSTM[18]	83.9	80.6
300D Tree-based CNN[19]		82.1
600D BiLSTM[20]	86.4	83.3
Deep Gated Attn. BiLSTM[22]	90.5	85.5
300D mLSTM[23]	92.0	86.1
BiMPM[24]	90.9	87.5
300D ESIM[9]	92.6	88.0
ESIM + ELMo[9]	91.6	88.7
LM-Pretrained Transformer[21]	96.6	89.9
$BERT_{base}$[12]	96.7	89.4
OSOA-DFN[27]	96.3	89.3
SemBERT[28]	96.4	89.8
Our	96.9	90.0

The training process employed an Early-Stop mechanism. After multiple rounds of training, when the validation accuracy showed a decreasing trend, the learning rate was halved. If, after Patience rounds of training, the validation accuracy didn't increase, the training was halted. The model parameters from the iteration with the highest validation accuracy were selected as the best model.

C Experimental Results and Analysis

Experimental Results Comparison: The experimental results of the recognition text entailment model based on word embeddings and attention on the SNLI dataset are compared with other models as shown in Table IV. To validate the effectiveness of the proposed method in recognition text entailment tasks, we compare it against classical models and representative model approaches that incorporate external resources. The first section includes

models based on simple neural networks, the second section consists of models integrating attention mechanisms, and the third section comprises models based on pre-trained language models or external resources.

It can be observed that the accuracy of simple neural network models like BiLSTM is around 83%. BiLSTM, built upon LSTM with bidirectional encoding, addresses LSTM's inability to capture information from both directions, making it more adept at handling sequence data. Meanwhile, the accuracy of the ESIM model, which incorporates attention mechanisms, reaches around 88%, approximately 5% higher than text encoding-based models. This improvement is attributed to the attention mechanism's ability to extract correlated information from sentences, capture inter-sentence relationships, and dynamically allocate weights to the output sentence information. ESIM model employs multiple reasoning steps to capture semantic relationships, strengthening the model's reasoning capacity through the attention mechanism.

Upon integrating pre-trained models, ESIM+ELMo exhibits a 0.7% increase in accuracy, indicating that pre-trained models can comprehend and learn deeper semantic information from the text. Pre-trained language models are capable of acquiring semantic representations at the sentence level.

From Table IV, it's evident that the recognition text entailment model based on Conformer and part-of-speech tagging integrates part-of-speech information into the attention mechanism. By allowing the model to learn the syntactic structure of input sentences based on part-of-

TABLE V. PERFORMANCE OF DIFFERENT MODELS ON THE MULTINLI DATASET

Model	match(%)	mismatch(%)
BiLSTM[20]	67.0	67.6
ESIM[9]	72.3	72.1
BiLSTM + ELMo +Attn[12]	76.4	76.1
KIM[10]	77.2	76.4
DIIN[25]	78.8	77.8
GPT[21]	82.1	81.4
$BERT_{base}$[12]	84.6	83.4
$BERT_{large}$[12]	86.7	85.9
Our	85.5	84.2

TABLE VI. ABLATION EXPERIMENT COMPARISON

Model	match(%)
ESIM	88.0
ESIM+ part of speech tagging	88.6
BERT	89.4
BERT+ part of speech tagging	89.8
Our	90.0

speech tagging information, the model's performance is further enhanced. This model achieves an accuracy of 90.0% on the SNLI dataset, surpassing BERT_base by 0.6%.

To validate the performance of this method across different datasets, the model was tested on the MultiNLI

dataset. The experimental results on the MultiNLI dataset are presented in Table V.

On the MultiNLI dataset, the recognition text entailment model based on Conformer and part-of-speech tagging achieves matching and mismatching accuracy rates of 85.5% and 84.2% respectively. These accuracy rates exceed those of the BERT_base model by 0.9% and 0.8% respectively. This demonstrates that the proposed model also exhibits excellent recognition performance across different datasets.

D Analysis of ablation experiment

As shown in Table VI, the recognition performance of ESIM, BERT, and conformer in textual entailment and the performance after adding post-tagging are shown. After integrating the information of part-of-speech tagging, the accuracy of ESIM and BERT increased by 0.6% and 0.4% respectively. Prove the effectiveness of integrating part-of-speech tagging on pre-trained language models for recognizing textual entailment tasks.In this experiment,N is 12. Through experiments, it was found that the model had the highest accuracy when the number of interaction module layers M=4 and the number of attention heads was 4,as shown in Figure 2.

V. CONCLUSION

This article proposes a recognition textual entailment model based on the Conformer and part-of-speech tagging, which builds on traditional models. The model uses the Conformer as a pre-trained language model, which combines the Transformer and Convolution models and is adept at processing long and complex sentences. On the basis of the Transformer model's ideology, it merges part-of-speech information and self-attention mechanisms, enhancing the ability of the self-attention mechanism to capture sentences, thereby enhancing the model's understanding of semantic relationships. The model's accuracy on the SNLI dataset reaches 90.0%, while on MultiNLI, the accuracy for matching and non-matching

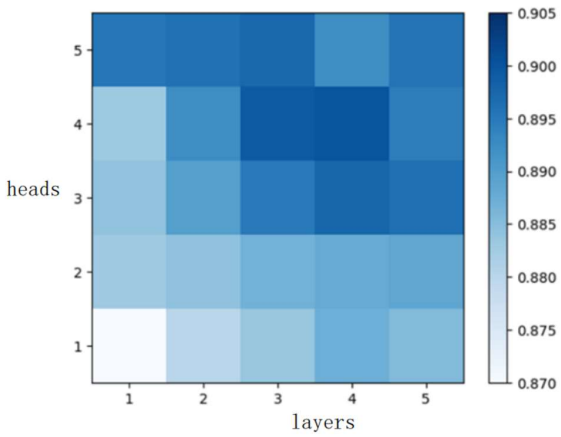

Fig.2. Compare the accuracy rates under different levels of interaction layers and head counts of multi-head attention.

pairs reaches 85.5% and 84.2% respectively. Finally, compared with different public models, its accuracy has been improved, demonstrating the effectiveness of the model and indicating that integrating part-of-speech information into large pre-trained models can enhance the performance of recognition textual entailment models.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Heilongjiang Province, grant LH2022F037.

REFERENCES

[1] Dagan, Ido, and Oren Glickman. "Probabilistic textual entailment: Generic applied modeling of language variability." Learning Methods for Text Understanding and Mining ,2004,pp.26-29.

[2] Lloret, Elena, et al. "A Text Summarization Approach under the Influence of Textual Entailment." NLPCS.,2008, pp. 22–31.

[3] Romano, Lorenza, et al. "Investigating a generic paraphrase-based approach for relation extraction." 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006), 2006,pp.409-416.

[4] Harabagiu, Sanda, and Andrew Hickl. "Methods for using textual entailment in open-domain question answering." Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,2006,pp.905–912.

[5] Dzikovska, Myroslava O., et al. "Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge." Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013),2013,pp.263–274.

[6] Jijkoun, Valentin, and Maarten de Rijke. "Recognizing textual entailment using lexical similarity." Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment,2005,pp. 73-76.

[7] Bowman, Samuel, Christopher Potts, and Christopher D. Manning. "Recursive neural networks can learn logical semantics." Proceedings of the 3rd workshop on continuous vector space models and their compositionality. 2015. pp.12-21.

[8] Rocktäschel, Tim, et al. "Reasoning about entailment with neural attention." arXiv preprint arXiv:1509.06664 (2015).

[9] Chen, Qian, et al. "Enhanced LSTM for natural language inference." arXiv preprint arXiv:1609.06038 (2016).

[10] Chen, Qian, et al. "Neural natural language inference models enhanced with external knowledge." arXiv preprint arXiv:1711.04289 (2017).

[11] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[12] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[13] Manning C D , Potts C , Angeli G ,et al.A large annotated corpus for learning natural language inference.2015.

[14] Williams, Adina , N. Nangia , and S. Bowman . "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference." 2018.

[15] Dozat, Timothy, and Christopher D. Manning. "Deep biaffine attention for neural dependency parsing." arXiv preprint arXiv:1611.01734 (2016).

[16] Ekbal, Asif, and Sriparna Saha. "Simulated annealing based classifier ensemble techniques: Application to part of speech tagging." Information Fusion 14.3 ,2013, pp.288－300.

[17] Gulati, Anmol, et al. "Conformer: Convolution-augmented transformer for speech recognition." arXiv preprint arXiv:2005.08100 (2020).

[18] Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv:1508.05326 ,2015.

[19] Mou, Lili, et al. "Natural language inference by tree-based convolution and heuristic matching." arXiv preprint arXiv:1512.08422 ,2015,.

[20] Liu, Yang, et al. "Learning natural language inference using bidirectional LSTM model and inner-attention." arXiv preprint arXiv:1605.09090 (2016).

[21] Radford A, Narasimhan K, Salimans T, et al. "Improving language understanding by generative pre-training". 2018.

[22] Chen, Qian, et al. "Recurrent neural network-based sentence encoder with gated attention for natural language inference." arXiv preprint arXiv:1708.01353 (2017).

[23] Wang, Shuohang, and Jing Jiang. "Learning natural language inference with LSTM." arXiv preprint arXiv:1512.08849 (2015).

[24] Wang, Zhiguo, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. The 26th International Joint Conference on Artificial Intelligence. Los Altos: William Kaufman, 2017.

[25] Gong, Y., Luo, H., & Zhang, J. (2017). Natural Language Inference over Interaction Space. *ArXiv, abs/1709.04348*.

[26] W. Zhu, T. Yao, W. Zhang and B. Wei, "Part-of-Speech-Based Long Short-Term Memory Network for Learning Sentence Representations," in IEEE Access, vol. 7, pp. 51810-51816, 2019.

[27] Liu, Mingtong, et al. "Original semantics-oriented attention and deep fusion network for sentence matching." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

[28] Zhang, Zhuosheng, et al. "Semantics-aware BERT for language understanding." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 05. 2020.