Natural Language Inference based on the LIC architecture with DCAE Feature

Jie Hu[1], Tanfeng Sun[1*], Xinghao Jiang[1], Lihong Yao[1], and Ke Xu[1]

[1] Shanghai Jiaotong University, Shanghai, China
{hujie_92781,tfsun,xhjiang,yaolh,xuke900708}@sjtu.edu.cn

Abstract. Natural Language Inference (NLI), which is also known as Recognizing Textual Entailment (RTE), aims to identify the logical relationship between a premise and a hypothesis. In this paper, a DCAE (Directly-Conditional-Attention-Encoding) feature based on Bi-LSTM and a new architecture named LIC (LSTM-Interaction-CNN) is proposed to deal with the NLI task. In the proposed algorithm, Bi-LSTM layers are used to modeling sentences to obtain a DCAE feature, then the DCAE feature is reconstructed into images through an interaction layer to enrich the relevant information and make it possible to be dealt with convolutional layers, finally the CNN layers are applied to extract high-level relevant features and relation patterns and the discriminant result obtained through a MLP (Multi-Layer Perceptron). Advantages of LSTM layers in sequence information processing and CNN layers in feature extraction are fully combined in this proposed algorithm. Experiments show this model achieving state-of-the-art results on the SNLI and Multi-NLI datasets.

Keywords: Natural Language Inference, Recognizing Textual Entailment, Attention, Bi-LSTM, Reconstructed Interaction, CNN.

1 Introduction

NLI, also known as RTE (Recognizing Textual Entailment), is a fundamental and challenging task in the field of NLP (Natural Language Processing). Textual Entailment is defined as the directed inference relationship between a pair of texts [1], if people believe that the semantic meaning of H (Hypothesis) can be inferred from the semantic meaning of P (Premise) according to their common sense, it is said that there is an entailment relation between P and H. The goal of RTE/NLI is to identify logical relationship (entailment, neutral, or contradiction) between a pair of P-H.

With the development of deep learning methods and large annotated datasets like SNLI [2] and Multi-NLI [3] was published, many researchers applied deep learning models to solve the NLI tasks and achieve lots of successful results [4-6].

The previous deep learning methods can be divide into two categories: 1) methods based on RNN models [5],[7-9], this kind of methods use RNN layers such as LSTM or GRU to encode premise and hypothesis separately, then simply concatenate the outputs of RNN layers as the feature of relation between the text pair, attention mechanism can also be used in the encoding phase in these methods; 2) methods based on interactive images or spaces [6],[10], this kind of methods convert the expressions of

* Corresponding Author

two sentences to an interactive image or space, then use image recognition ways to solve textual relation identification problem.

Researchers using methods in the first category believe that the structure of RNN is better suited for processing temporal information such as texts, while too little interactive information is considered in their framework. Indeed, attention mechanism can obtain some interactive features to help optimize those RNN models, but the effect is limited. And researchers using methods in the second category preserve most of the interactive information between texts, while some temporal features are alleviated in the process.

In this paper, a new DCAE feature based on Bi-LSTM and a new architecture named LIC (LSTM-Interaction-CNN) is proposed to solve the NLI task. The DCAE feature fuses three mainstream encoding methods include directly Bi-LSTM encoding, conditional encoding and encoding with attention. In the LIC architecture, Bi-LSTM layers are used to modeling sentences since their design characteristics are very suitable for the modeling of sequential data, then the encoding vectors are reconstructed into images through an interaction layer to enrich the relevant information between inputs and make it possible to be dealt with convolutional layers, finally the CNN layers are applied to extract high-level relevant features and relation patterns. Advantages of LSTM layers in sequence information processing and CNN layers in feature extraction are fully combined in this proposed LIC architecture. And based on the proposed DCAE feature and the LIC architecture, a complete algorithm for the NLI task was built with a MLP to obtain the relationship label.

The main contributions of this paper include: 1) propose a new DCAE (Directly-Conditional-Attention-Encoding) feature based on multi-features obtained by different Bi-LSTM encoding to incorporate these complementary encoding methods 2) propose a LIC (LSTM-Interaction-CNN) architecture fully combined the advantages of LSTM layers in sequence information processing and CNN layers in feature extraction to deal with the NLI task.

2 Proposed Algorithm

2.1 Directly-Conditional-Attention-Encoding Feature

LSTM (Long-Short-Term Memory), which is known as a kind of RNN (Recurrent Neural Network), is very suitable for the modeling of sequential data, such as text data due to its design characteristics. Bi-LSTM is a variant of LSTM, which is composed of forward directional LSTM and backward directional LSTM and is often used in NLP tasks to model context information. While in the actual sentence modeling phase, there are some different choices such as directly Bi-LSTM encoding [2], conditional encoding [11] and encoding with attention features [12] and so on. All of these methods were demonstrated to be effective in some specific NLP tasks. In this paper a DCAE feature based on all of these Bi-LSTM encoding methods is proposed, experiments in section 3 show the effectiveness of the DCAE feature. And then, how to obtain the DCAE feature will be elaborated.

Directly Bi-LSTM Encoding. Bi-LSTM layers are used to directly encoding the texts at first, and the process of a traditional Bi-LSTM encoding an input is illustrated as Figure 1.

In NLI task, an input includes a pair of premise and hypothesis, then they are encoded to:

$$(\bar{p}, h_p) = BiLSTM(p, 0) \tag{1}$$

$$(\bar{h}, h_h) = BiLSTM(h, 0) \tag{2}$$

Corresponding to Figure 1, the compositions of $(\bar{p} \in \mathbb{R}^{n\times 2d}, h_p \in \mathbb{R}^{2d})$ are:

$$(\bar{p}, h_p) = ([hf_1 + hb_1, hf_2 + hb_2, hf_3 + hb_3, \dots, hf_n + hb_n], (hf_n, hb_n)) \quad (3)$$

The compositions of $(\bar{h}, h_h)$ are similar to $(\bar{p}, h_p)$. And it's worthy to mention that the Bi-LSTM encoding premise and the Bi-LSTM encoding hypothesis shared weight matrices while training.

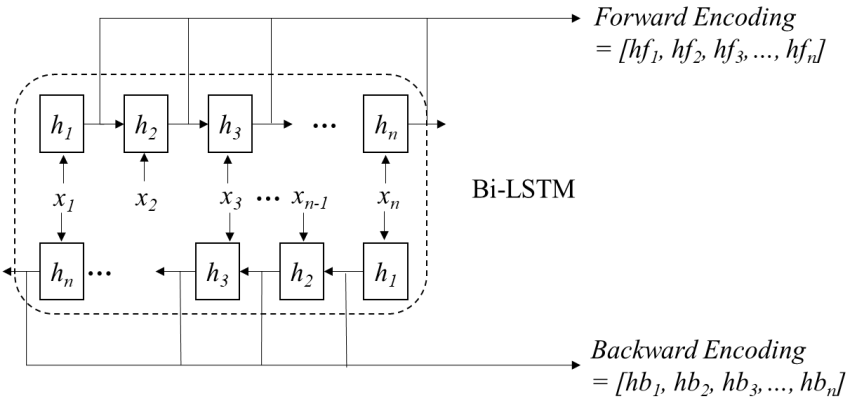

Fig.1 Process of a Bi-LSTM encoding an input

Conditional Encoding. Conditional encoding is led to obtaining some relation features between premise and hypothesis then. It was inspired by [11]. In this section, the final hidden state from the Bi-LSTM which encoded the premise is used to initialize the Bi-LSTM which will encode the hypothesis, and vice versa. Then some relevant information about the premise and hypothesis was led to the encoding results.

To conditional encode premise and hypothesis, the outputs from directly Bi-LSTM encoding will be used. Conditional encoding results are obtained as following:

$$(\tilde{p}, -) = BiLSTM(p, h_h) \quad (4)$$

$$(\tilde{h}, -) = BiLSTM(h, h_p) \quad (5)$$

where $-$ means the value generated here is not in consideration, $\tilde{p} \in \mathbb{R}^{n\times 2d}, \tilde{h} \in \mathbb{R}^{m\times 2d}$.

Attention Encoding. A soft alignment attention mechanism was introduced into this section to obtain more relevant information between the original texts. Here an unconventional attention weight is computed to express the similarity of hidden states of the premise and the hypothesis [5]:

$$e_{ij} = \tilde{p}_i \tilde{h}_j^T, \qquad i \in [1, n], j \in [1, m] \quad (6)$$

where *n*, *m* respectively are the length of premise and hypothesis, and $\tilde{p}_i \in \mathbb{R}^{2d}$, $\tilde{h}_j \in \mathbb{R}^{2d}$ are the components of $\tilde{p}$, $\tilde{h}$, which are the results from conditional encoding. The unconventional attention weight is more simple than conventional attention weight but still effective due to its "soft alignment" by using corresponding word vectors from premise and hypothesis to do the dot product operation. And the final attention encoding results $\hat{p}_i \in \mathbb{R}^{n\times 2d}, \hat{h}_i \in \mathbb{R}^{m\times 2d}$ are obtained as following:

$$\hat{p}_i = \sum_{j=1}^{l_p} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_h} \exp(e_{ik})} \tilde{h}_j, \qquad i \in [1, n] \quad (7)$$

$$\widehat{h_\iota} = \sum_{i=1}^{l_h} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_p} \exp(e_{ik})} \widetilde{p_\iota}, \qquad j \in [1, m] \tag{8}$$

The DCAE Feature. After the above three steps of encoding, the directly Bi-LSTM encoding vectors $(\bar{p}, \bar{h})$, the conditional encoding vectors $(\tilde{p}, \tilde{h})$ and the attention encoding vectors $(\hat{p}, \hat{h})$ were obtained. Experiments on individual feature will show in section 3. Then how to fuse these separate features into an effective composite feature should be considered.

Common methods of fusion between two vectors include element-wise subtraction, element-wise production and so on. Element-wise subtraction is usually applied to extrude the different part between the two features, while element-wise production is applied to reinforce the relevant parts of the two features.

In the situation mentioned in this paper, the directly Bi-LSTM encoding vectors just contain the information of a single sentence, while conditional encoding and the attention encoding are both based on a pair of texts and contain various degrees of relevant information. Then element-wise subtraction can be applied to extract the difference between the directly Bi-LSTM vectors and the conditional vectors, and the difference between the directly Bi-LSTM vectors and the attention vectors to maximum the retention of relevant information and remove the redundancy. And the element-wise production can be applied to reinforce the relevant part of the conditional vectors and the attention vectors to maintain and strengthen the relevant information.

And then a DCAE feature based on these three encoding features is proposed:

$$\overrightarrow{p_\iota} = [\bar{p_\iota}, \widetilde{p_\iota} - \bar{p_\iota}, \widehat{p_\iota} - \bar{p_\iota}, \widetilde{p_\iota} \odot \widehat{p_\iota}] \tag{9}$$

$$\overrightarrow{h_j} = [\bar{h_j}, \widetilde{h_j} - \bar{h_j}, \widehat{h_j} - \bar{h_j}, \widetilde{h_j} \odot \widehat{h_j}] \tag{10}$$

where $-$ represents element-wise subtraction, and $\odot$ means element-wise multiplication. And the whole process of how to obtain the DCAE feature proposed in this paper is illustrated in Figure 2.

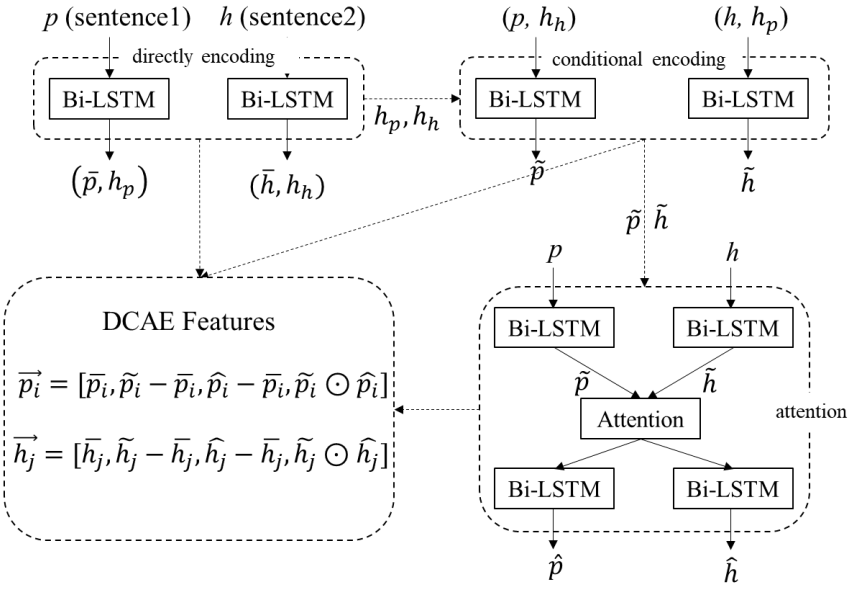

Fig.2 The process of obtaining the DCAE features

2.2 LSTM-Interaction-CNN Architecture

The proposed LIC architecture contains three main parts: the LSTM encoding part, the reconstructed interaction part, and the CNN feature extraction part, aiming to establish an effective model for classification or discrimination tasks of a pair of texts or other coupled temporal input. The superiority of the LIC is that advantages of LSTM layers in sequence information processing and CNN layers in feature extraction are fully combined in this architecture. In a LIC, LSTM layers are used to modeling sentences to feature vectors, then the feature vectors are reconstructed into images through an interaction layer to enrich the relevant information and make it possible to be dealt with convolutional layers, finally the CNN layers are applied to extract high-level features for subsequent classification or discrimination tasks. And then how to apply the LIC architecture to NLI task will be elaborated.

Bi-LSTM Encoding Layer. In this phase, a pair of premise and the corresponding hypothesis will be encoding to be feature vectors. And in this paper, the P-H pair are encoded to two 4-dimentional feature vectors (the DCAE feature proposed in this paper) shown in section 2.1.

Reconstructed Interaction Layer. In this phase, DCAE features of the premise and the corresponding hypothesis will be converted into images to enrich the interactive information and make it possible to extract high level features by convolutional layers. It was mentioned in section 2.1, a DCAE feature contains four elements, for each element of DCAE features for a pair of P-H, an interactive image will generate.

To illustrate this process more specifically, directly encoding vectors are used to be an example, and the process of interaction shows in Figure 3.

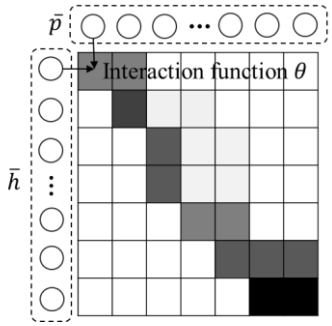

Fig.3 Directly encoding results of P-H generate an interactive image

M is used to represent the interactive image, then:

$$M = f(\theta; \bar{p}, \bar{h}) \tag{11}$$

$$m_{ij} = \theta(\bar{p}_i, \bar{h}_j) \tag{12}$$

where θ represents the interaction function, and in this paper, dot product was chosen to be θ.

After the reconstructed interaction layer, four images were generated corresponding to a pair of premise and hypothesis due to the DCAE feature contains four elements.

Convolutional Layer. Convolutional layers are implicated to extract high-level relevant information features and relation patterns, and in this paper, a two-layers CNN model is used to extract hierarchical

features, since the first layer was supposed to obtain phase-level interaction patterns, and the second layer was supposed to obtain sentence-level interaction patterns.

The framework of how this two-layer CNN model works showed in the Figure 4. It's worth mentioning that activation function ReLU was introduced to activate feature maps generated in the process. Then the entire feature extraction process can be expressed as:

$$M_i^1 = \sum_{k=0}^{c-1} weigt(M_i^1, k) \otimes M_k^0 + bias(M_i^1) \tag{13}$$

$$N^1 = maxpooling[ReLU(M^1)] \tag{14}$$

$$M_i^2 = \sum_{k=0}^{c-1} weigt(M_i^2, k) \otimes N_k^1 + bias(M_i^2) \tag{15}$$

$$N^2 = maxpooling[ReLU(M^2)] \tag{16}$$

where M^0 is the multi-channel interactive image obtained in last section, c (which should be 4 since 4 interactive images were generated for each P-H) is the number of channels, M^* represent the outputs of convolutional layers, N^* represent the outputs of max-pooling layers, and weight, bias are weight matrices to be learned.

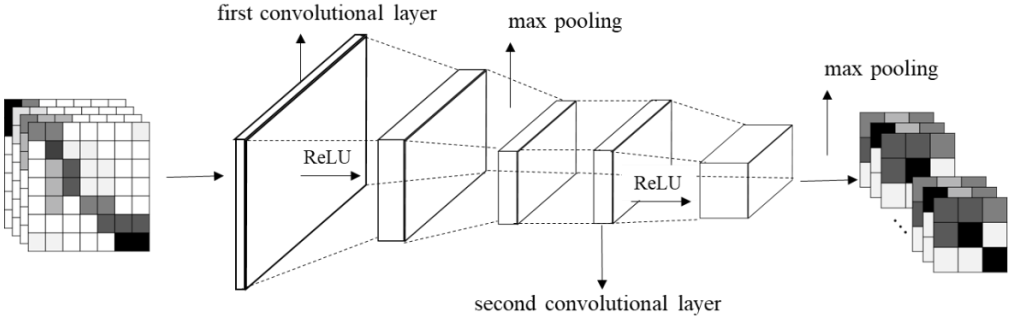

Fig.4 The two-layer CNN model extract hierarchical features

3 Framework Of The Proposed Algorithm

In this section, the complete framework to solve the NLI task based on the proposed algorithms is presented and can be illustrated in Figure 5. In addition to the LIC architecture proposed in section 2.2, MLP (Multi-Layer Perception) is also introduced into the framework to obtain the final discriminant result. Then there are four main steps: DCAE Feature Obtainment, Reconstructed Interaction, CNN Feature Extraction and the MLP Prediction.

Step 1. DCAE Feature Obtainment. In this step, a pair of premise and the corresponding hypothesis are encoded to two 4-dimentional DCAE feature vectors shown in section 2.1.

Step 2. Reconstructed Interaction. In this step, DCAE features of the premise and the corresponding hypothesis are converted into images to enrich the interactive information and make it possible to extract high-level features by convolutional layers. And four images were generated corresponding to a pair of premise and hypothesis since the DCAE feature contains four elements.

Step 3.CNN Feature Extraction. In this step, a two-layers CNN model is used to extract hierarchical relevant information features and matching patterns, since the first layer was supposed to obtain phase-level matching patterns, and the second layer was supposed to obtain sentence-level matching patterns.

Step 4. MLP Prediction. Several interactive feature maps contain phase-level and sentence-level patterns were obtained after the process of convolution layers, and in this step, the final discriminant result based on those feature maps will be obtained through an MLP (Multi-Layer Perception):

$$Result = MLP(N^2) \tag{17}$$

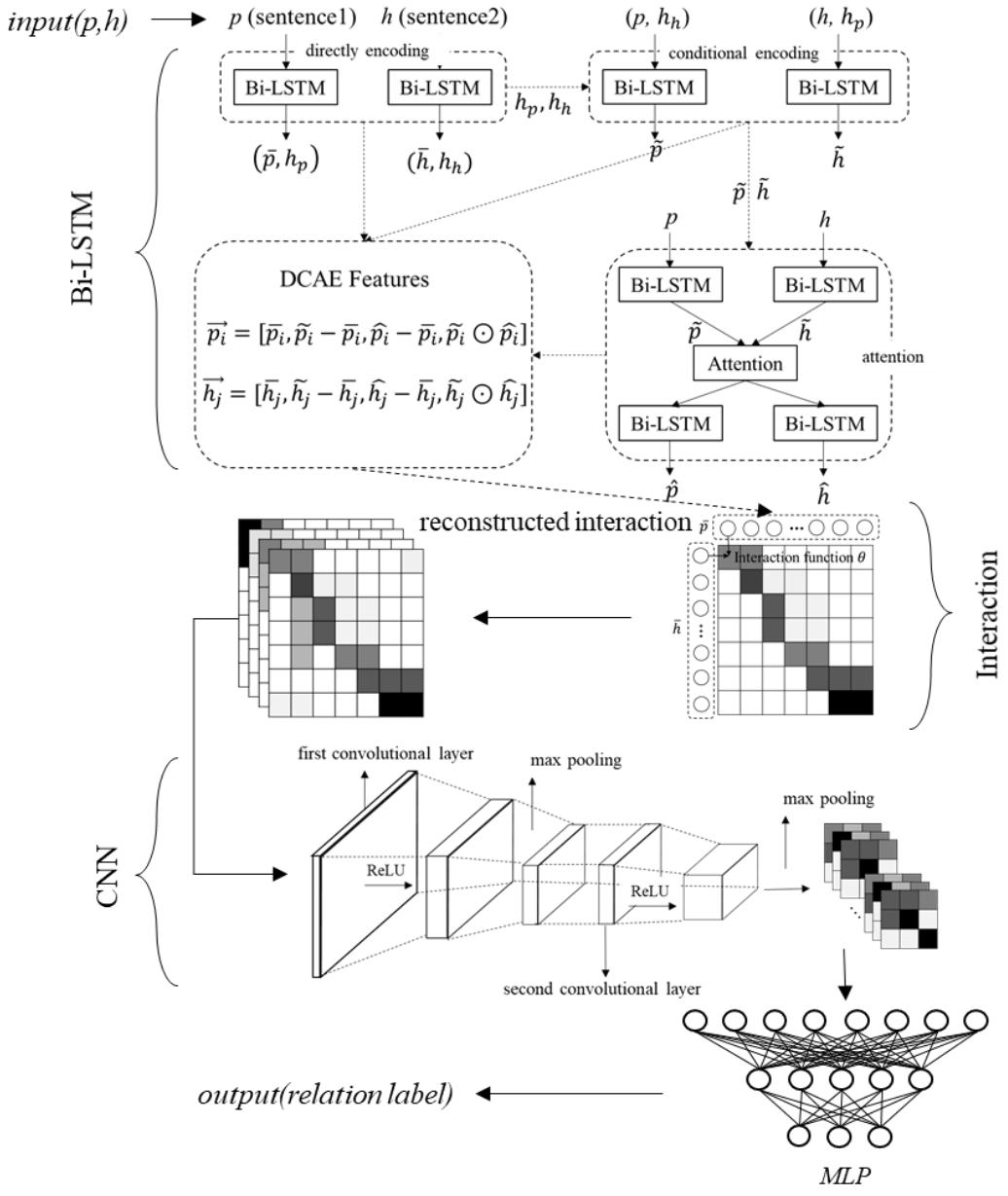

Fig.5 Framework to solve NLI task based on the proposed algorithm

4 Experiments and Analysis

4.1 Setup

Data. In this paper, two datasets focused on NLI task were evaluated. They are SNLI (Stanford Natural Language Inference Corpus) dataset [2] and the Multi-NLI (Multi-Genre Natural Language Inference Corpus) dataset [3].

The SNLI dataset contains 570K sentence pairs with human annotated as entailment, neutral, contradiction or -, where - indicates a lack of consensus from human annotators. The premises were collected from the Flickr30k corpus [13], and the hypotheses were then manually adapted corresponding to each relation type.

The Multi-NLI, which considered to be an upgrade version for the SNLI, has 433K sentence pairs with similar collection process as SNLI. The difference is the premises were collected from maximally broad range of genre of American English included written non-fiction genres, spoken genres, less formal written genres and a specialized one for 9/11. And even more subtly, half of these selected genres appear in training set while another half are not, creating in domain (matched) and cross-domain (mismatched) test sets to validate the generalization capabilities of models.

Training. The model proposed in this paper is implemented with Pytorch framework. Pre-trained 300-D Glove 840B vectors [14] are used to initialize the word embedding vectors from the original texts and the vectors are updated during training. Two layers of Bi-LSTM are applied to directly encoding and conditional encoding while two attention units (three layers Bi-LSTM totally) are applied to encode sentences with attention. Then two layers of convolutional layers are used to obtain phase-level and sentence-level patterns, while kernel size is set to be (5,5) and (3,3). And finally, two fully connected layers are used to generate the discriminant result. The initial learning rate is set to be 0.0004 during the training, and dropout [15] with a rate of 0.4 for regularization is applied to all feedforward connections to avoid overfitting.

4.2 Compare With Other Works

Table 1 shows the accuracy of the models on test sets of SNLI and Multi-NLI. Result in the first row is a baseline classifier presented by Bowman et al. (2015) [2], who is the founder of the SNLI dataset and utilizes handcrafted features. Deep learning methods are used in all of the rest models listed in the table 1, then their effectiveness can be seen from the accuracy gap between the traditional methods and deep learning methods.

And as is shown in the table 1, the proposed model achieves competitive results between the deep learning models. The first three models are based on Bi-LSTM. Gated-Att BiLSTM (Chen Q et al.,2017) [16] uses Bi-LSTM with gated attention feature to encode the sentence pairs, then obtained the result by a MLP. DR-BiLSTM (Reza G et al.,2018) [5] uses multi-features include dependent encoding and encoding with attention which are the same as conditional encoding and encoding with attention in this paper, then uses another Bi-LSTM for inference and named this mechanism as Deep Reading. And HIM (Hybrid Inference Model) (Qian Chen et al.,2017) [9] model consists ESIM (Enhanced Sequential Inference Model) which also uses multi-level features based on Bi-LSTM to encode sentences and a tree-LSTM model to collect local inference information and obtain the result.

The biggest difference between this paper and those methods that separate the two processes of feature extraction and relationship inference like the DR-BiLSTM and the HIM is that CNN is used to

extract high-level relation patterns in the proposed model since CNN structure has advantages in image feature extraction, while Bi-LSTM is used in the inference phase in DR-BiLSTM model to maintain the continuity of feature vectors and a tree-LSTM structure is used in the HIM model to collect local inference information. Experiments show that the proposed model is 0.5 percentage points more accurate than the DR-BiLSTM and 0.4 percentage points more accurate than the HIM when test on the SNLI dataset.

DIIN (Densely Interactive Inference Network) (Yichen Gong et al.,2018) [6] model is based on interaction space, which uses pre-trained embedding vectors and other statistical characteristic like POS (one-hot part-of speech) and EM (binary exact match) features to generate an interaction space, then use a CNN-based model to extract relation features to obtain the final result. Experiments show that the proposed model is 1.0 percentage point more accurate than DIIN when test on the SNLI dataset, and 0.2 percentage points on Matched Multi-NLI, 0.7 percentage points on Mismatched Multi-NLI.

And the DMAN (Discourse Marker Augmented Network) (Boyuan Pan et al.,2018) [17] mainly uses reinforcement learning mechanism (distinguished from the proposed algorithms but still belongs to deep learning methods) to deal with the NLI task, while the proposed algorithm is still 0.2 percentage points more accurate than DMAN on average.

Table 1. Accuracy of the models on the test sets of SNLI and Multi-NLI

Model	**SNLI**	**Multi-NLI**	
		Matched	Mismatched
Handcrafted features (Bowman et al.,2015) [2]	78.2%	-	-
Gated-Att BiLSTM (Chen Q et al.,2017) [16]	85.5%	76.8%	75.8%
DR-BiLSTM (Single) (Reza G et al.,2018) [5]	88.5%	-	-
HIM (Qian Chen et al.,2017) [9]	88.6%	-	-
DIIN (Single) (Yichen Gong et al.,2018) [6]	88.0%	78.8%	77.8%
DMAN (Single) (Boyuan Pan et al.,2018) [17]	88.8%	78.9%	78.2%
DCAE Feature + LIC Architecture	**89.0%**	**79.0%**	**78.5%**

In a nutshell, the proposed model achieves better results than both models based on Bi-LSTM and models based on interactive image or space and achieves competitive results when compares to other deep learning methods.

4.3 Performance Experiments

Table 2 shows the accuracy of the different features or parts of the proposed algorithm.

Attention Bi-LSTM is supposed to be the best single feature for sentence encoding in NLI task based on the experiments, and the DCAE feature proposed in this paper is 1.2 percentage points more accurate than the best single feature. The addition of DCAE features improves the accuracy by 14.4 percentage points on the SNLI dataset since the model use Pre-trained vectors (Pre-trained 300-D Glove 840B vectors [14], are considered to contain no relation features) and the LIC architecture achieves an accuracy of 74.6% while the model use DCAE feature and LIC architecture achieves an accuracy of 89.0%. On the Multi-NLI dataset, the addition of DCAE feature even improves the accuracy by nearly

17 percentage points. By the same token, it can be seen from the table 2 that the addition of LIC architecture improves the accuracy by 2.7 percentage points on SNLI dataset and nearly 2.5 percentage points on Multi-NLI dataset.

Table 2. Accuracy of different encoding methods and architecture

Model	SNLI	Multi-NLI	
		Matched	Mismatched
Directly Bi-LSTM + SVM	80.6%	69.7%	68.6%
Conditional Bi-LSTM + SVM	83.2%	72.4%	71.8%
Attention Bi-LSTM +SVM	85.1%	74.6%	74.3%
DCAE Feature + SVM	86.3%	76.6%	76.0%
Pre-trained vectors + LIC Architecture	74.6%	62.4%	61.5%
DCAE Feature + LIC Architecture	**89.0%**	**79.0%**	**78.5%**

It can be learned from these data that the DCAE feature does incorporate three complementary encoding methods, and the LIC architecture can also improve the effectiveness of the model.

5 Conclusion

A new DCAE feature based on Bi-LSTM and a new architecture named LIC (LSTM-Interaction-CNN) is proposed to deal with the NLI task in this paper. The DCAE feature fuses three mainstream encoding methods include directly Bi-LSTM encoding, conditional encoding and encoding with attention. And in the algorithm based on the LIC architecture, Bi-LSTM layers are used to modeling sentences to obtain the DCAE feature, then the DCAE features are reconstructed into images through an interaction layer to enrich the relevant information and make it possible to be dealt with convolutional layers, finally the CNN layers are applied to extract high-level relevant features and relation patterns and the discriminant result obtained through a MLP. The advantages of Bi-LSTM layers in sequence information processing and the advantages of CNN layers in feature extraction are fully combined in the proposed algorithm. And experiments show the proposed algorithm achieves better results than models based on Bi-LSTM or interactive images and even other deep learning methods.

Acknowledgements

This work is funded by National Key Research and Development Projects of China（2018YFC0830703）. It is also supported by National Natural Science Foundation of China (Grant No.61572320 & 61572321). The corresponding author is Dr. Tanfeng Sun.

6 References

1. Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. Learning Methods for Text Understanding and Mining 2004, 26–29 (2004)
2. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 632–642 (2015)

3. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1112–1122 (2018)
4. Tay, Y., Luu, A.T., Hui, S.C.: Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1565–1575 (2018)
5. Ghaeini, R., Hasan, S.A., Datla, V., Liu, J., Lee, K., Qadir, A., Ling, Y., Prakash, A., Fern, X., Farri, O.: Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1460–1469 (2018)
6. Gong, Y., Luo, H., Zhang, J.: Natural language inference over interaction space. arXiv preprint arXiv:1709.04348 (2017)
7. Liu, Y., Sun, C., Lin, L., Wang, X.: Learning natural language inference using bidirectional lstm model and inner-attention. arXiv preprint arXiv:1605.09090 (2016)
8. Wang, S., Jiang, J.: Learning natural language inference with lstm. In: Proceedings of NAACL-HLT. pp. 1442–1451 (2016)
9. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1657– 1668 (2017)
10. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2793–2799. AAAI Press (2016)
11. Rockt¨aschel, T., Grefenstette, E., Hermann, K.M., Ko˘cisky`, T., Blunsom, P.: Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664 (2015)
12. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 551–561 (2016)
13. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015)
14. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
16. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Recurrent neural network-based sentence encoder with gated attention for natural language inference. In: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP. pp. 36–40 (2017)
17. Pan, B., Yang, Y., Zhao, Z., Zhuang, Y., Cai, D., He, X.: Discourse marker augmented network with reinforcement learning for natural language inference. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 989–999 (2018)