Making the Relation Matters: Relation of Relation Learning Network for Sentence Semantic Matching

Kun Zhang[†], **Le Wu**[†‡], **Guangyi Lv**[§], **Meng Wang**[†‡*], **Enhong Chen**[§], **Shulan Ruan**[§]
[†]School of Computer Science and Information Engineering, Hefei University of Technology
[‡] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[§]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
{zhang1028kun, lewu.ustc, eric.mengwang}@gmail.com, {gylv,slruan}@mail.ustc.edu.cn, cheneh@ustc.edu.cn

Abstract

Sentence semantic matching is one of the fundamental tasks in natural language processing, which requires an agent to determine the semantic relation among input sentences. Recently, deep neural networks have achieved impressive performance in this area, especially BERT. Despite their effectiveness, most of these models treat output labels as meaningless one-hot vectors, underestimating the semantic information and guidance of relations that these labels reveal, especially for tasks with a small number of labels. To address this problem, we propose a *Relation of Relation Learning Network (R^2-Net)* for sentence semantic matching. Specifically, we first employ BERT to encode the input sentences from a global perspective. Then a CNN-based encoder is designed to capture keywords and phrase information from a local perspective. To fully leverage labels for better relation information extraction, we introduce a self-supervised relation of relation classification task for guiding R^2-*Net* to consider more about relations. Meanwhile, a triplet loss is employed to distinguish the intra-class and inter-class relations in a finer granularity. Empirical experiments on two sentence semantic matching tasks demonstrate the superiority of our proposed model. As a byproduct, we have released the codes to facilitate other researches.

1 Introduction

Sentence semantic matching is a fundamental *Natural Language Processing (NLP)* task that tries to infer the most suitable label for a given sentence pair. For example, Natural Language Inference (NLI) targets at classifying the input sentence pair into one of the three relations (i.e., *Entailment, Contradiction, Neutral*) (Kim et al. 2018). Paraphrase Identification (PI) aims at identifying whether the input sentence pair expresses the same meaning (Dolan and Brockett 2005). Figure 1 gives some examples with different semantic relations from different datasets.

As a fundamental technology, sentence semantic matching has been applied successfully into many NLP fields, e.g., information retrieval (Clark et al. 2016), question answering (Liu et al. 2018), and dialog system (Serban et al. 2016). Currently, most work leverages the advancement of representation learning techniques (Devlin et al. 2018; Vaswani et al. 2017) to tackle this task. They focus on input sentences and design different architectures to explore sentence semantics comprehensively and precisely. Among all these methods, BERT (Devlin et al. 2018) plays an important role. It adopts multi-layer transformers to make full use of large corpus (i.e., BooksCorpus and English Wikipedia) for the powerful pre-trained model. Meanwhile, two self-supervised learning tasks (i.e., Masked LM and Next Sentence Prediction) are designed to better analyze sentence semantics and capture as much information as possible. Based on BERT, plenty of work has made a big step in sentence semantic modeling (Liu et al. 2019; Radford et al. 2018).

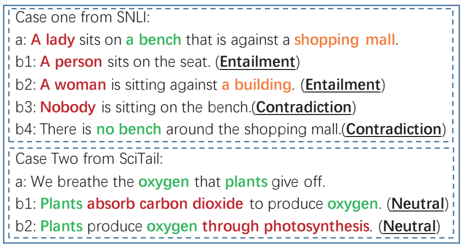

Case one from SNLI:
a: A lady sits on a bench that is against a shopping mall.
b1: A person sits on the seat. (**Entailment**)
b2: A woman is sitting against a building. (**Entailment**)
b3: Nobody is sitting on the bench.(**Contradiction**)
b4: There is no bench around the shopping mall.(**Contradiction**)
Case Two from SciTail:
a: We breathe the oxygen that plants give off.
b1: Plants absorb carbon dioxide to produce oxygen. (**Neutral**)
b2: Plants produce oxygen through photosynthesis. (**Neutral**)

Figure 1: Some examples from SNLI and SciTail datasets.

In fact, since relations are the predicting targets of sentence semantic matching task, most methods do not pay enough attention to the relation learning. They just leverage annotated labels to represent relations, which are formulated as one-hot vectors. However, these independent and meaningless one-hot vectors cannot reveal the rich semantic information and guidance of relations (Zhang et al. 2018), which will cause an information loss. Gururangan et al. (2018) has observed that different relations among sentence pairs imply specific semantic expressions. Taking Figure 1 as an example, most sentence pairs with "*contradiction*" relation contain negation words (e.g., *nobody, never*). "*entailment*" relation often leads to exact numbers being replaced with approximates (*person, some*). "*Neutral*" relation will import some correct but irrelevant information (e.g., *absorb carbon dioxide*). Moreover, the expressions between sentence pairs with different relations are very different. Therefore, the comparison and contrastive learning among different relations (e.g., pairwise relation learning) can help models to learn more about the semantic information implied in the re-

[*]Corresponding author

lations, which in turn helps to strengthen the sentence analysis ability of models. They should be treated as more than just meaningless one-hot vectors.

One of the solutions for better relation utilization is the embedding method inspired by Word2Vec. Some researchers try to jointly encode the input sentences and labels in the same embedding space for better relation utilization during sentence semantic modeling (Du et al. 2019; Wang et al. 2018a). Despite the progress they have achieved, label embedding method requires more data and parameters to achieve better utilization of relation information. It still cannot fully explore the potential of relations due to the small number of relation categories or the lack of explicit label embedding initialization (Wang et al. 2018a).

To this end, in this paper, we propose a novel *Relation of Relation Learning Network (R^2-Net)* approach to make full use of relation information in a simple but effective way. In concrete details, we first utilize pre-trained BERT to model semantic meanings of the input words and sentences from a global perspective. Then, we develop a CNN-based encoder to obtain partial information (*keywords and phrase information*) of sentences from a local perspective. Next, inspired by self-supervised learning methods in BERT training processing, we propose a **R**elation of **R**elation (R^2) classification task to enhance the learning ability of *R^2-Net* for the implicit common features corresponding to different relations. Moreover, a triplet loss is used to constrain the model, so that the intra-class and inter-class relations are analyzed better. Along this line, input sentence pairs with the same relations will be represented much closer and vice versa further apart. Relation information is properly integrated into sentence pair modeling processing, which is in favor of tackling the above challenges and improving the model performance. Extensive evaluations of two sentence semantic matching tasks (i.e., NLI and PI) demonstrate the effectiveness of our proposed *R^2-Net* and its advantages over state-of-the-art sentence semantic matching baselines.

2 Related Work

In this section, we mainly introduce the related work from two aspects: 1) *Sentence Semantic Matching*, and 2) *Label Embedding for Text Classification*.

2.1 Sentence Semantic Matching

With the development of various neural network technologies such as CNN (Kim 2014), GRU (Chung et al. 2014), and the growing importance of the attention mechanism (Vaswani et al. 2017; Parikh et al. 2016), plenty of methods have been exploited for sentence semantic matching on large datasets like SNLI (Bowman et al. 2015), SciTail (Khot et al. 2018), and Quora (Iyer et al. 2017). Traditionally, researchers try to fully use neural network technologies to model semantic meanings of sentences in an end-to-end fashion. Among them, CNNs focus on the local context extraction with different kernels, and RNNs are mainly utilized to capture the sequential information and semantic dependency. For example, Mou et al. (2016a) employed a tree-based CNN to capture the local context information in sentences. Kun et al. (2018) combined CNN and GRU into a hybrid architecture, which utilizes the advantages of both networks. They used CNN to generate phrase-level semantic meanings and GRU to model the word sequence and dependency between sentences.

Recently, attention-based methods have shown very promising results on many NLP tasks, such as machine translation (Bahdanau et al. 2014), reading comprehension (Zheng et al. 2019), and NLI (Bowman et al. 2016). Attention helps to extract the most important parts in sentences, capture semantic relations, and align the elements of two sentences properly (Cho, Courville, and Bengio 2015; Zhang et al. 2017). It has become an essential component for improving model performance and sentence understanding. Early attempts focus on designing different attention methods that are suitable for specific tasks, like inner-attention (Liu et al. 2016), co-attention (Kim et al. 2018), and multi-head attention (Shen et al. 2017). To fully explore the potential of attention mechanism, Zhang et al. (2019) proposed a dynamic attention mechanism, which imitates human reading behaviors to select the most important word at each reading step. This method has achieved impressive performance. Another direction is pre-trained methods. Devlin et al. (2018) used very large corpus and multi-layer transformers to obtain a powerful per-trained BERT. This method leverages multi-head self-attention to encode sentences and achieves remarkable performances on various NLP tasks. With the powerful representation ability, pre-trained BERT model has accelerated the NLP research.

However, most of these methods only focus on the input sentences and treat the labels as meaningless one-hot vectors, which ignores the potential of label information (Zhang et al. 2018). There still remains plenty of space for further improvement on sentence semantic matching.

2.2 Label Embedding for Text Classification

As an extremely important part of training data, labels contain much implicit information that needs to be explored. In computer vision, researchers have proposed label embedding methods to make full use of label information.

However, research on explicit label utilization in NLP is still a relatively new domain. One possible reason is that there are not that many labels in NLP tasks. Thus, label information utilization is only considered on the task with relatively a large number of labels or multi-task learning. For example, Zhang et al. (2018) proposed a multi-task label embedding method for better implicit correlations and common feature extraction among related tasks. Du et al. (2019) designed an explicit interaction model to analyze the fine-grained interaction between word representations and label embedding. They have achieved impressive performance on text classification tasks. In addition, Wang et al. (2018a) and Pappas and Henderson (2019) transferred the text classification task to a label-word joint embedding problem. They leveraged the semantic vectors of labels to guide models to select the important and relevant parts of input sentences for better performance. The above work demonstrates the superiority of explicit label utilization and inspires us to make better use of label information.

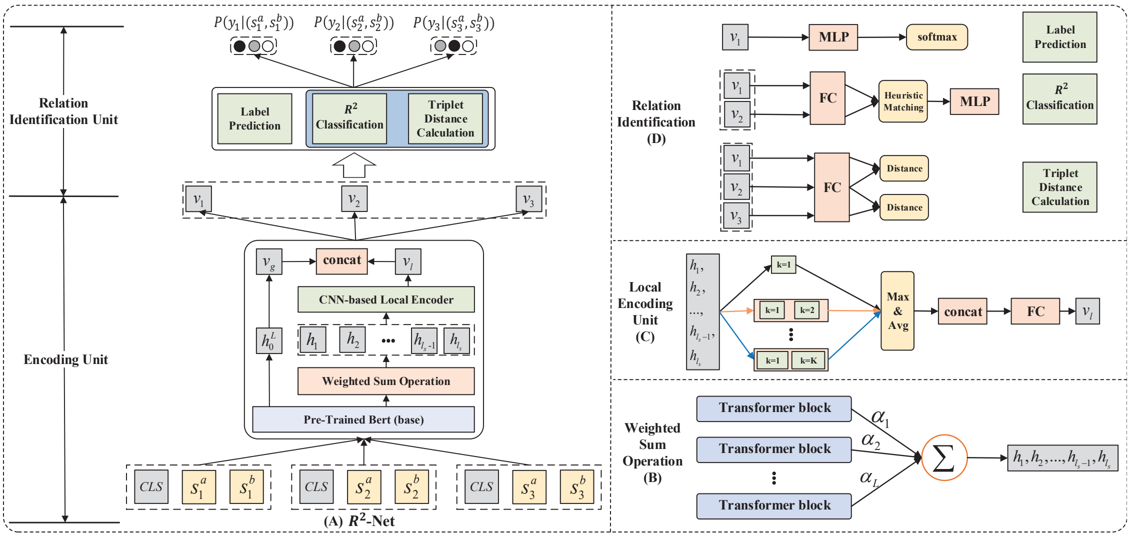


Figure 2: Architecture of *Relation of Relation Learning Network (*R^2*-Net)*.

3 Problem Statements

In this section, we will introduce the definition of sentence semantic matching task and our proposed relation of relation classification task.

3.1 Sentence Semantic Matching

Sentence semantic matching task can be formulated as a supervised classification. Given two input sentences $\boldsymbol{s}^a = \{\boldsymbol{x}_1^a, \boldsymbol{x}_2^a, ..., \boldsymbol{x}_{l_a}^a\}$ and $\boldsymbol{s}^b = \{\boldsymbol{x}_1^b, \boldsymbol{x}_2^b, ..., \boldsymbol{x}_{l_b}^b\}$, where $\boldsymbol{x}_i^a$ and $\boldsymbol{x}_j^b$ are feature tokens for each sentence. The goal of this task is to train a classifier ξ, which is capable of computing the conditional probability $P(y|\boldsymbol{s}^a, \boldsymbol{s}^b)$ and predicting the relation for input sentence pair based on the probability.

$$\begin{aligned} &P(y|\boldsymbol{s}^a, \boldsymbol{s}^b) = \xi(\boldsymbol{s}^a, \boldsymbol{s}^b), \\ &y^* = argmax_{y \in \mathcal{Y}} P(y|\boldsymbol{s}^a, \boldsymbol{s}^b), \end{aligned} \tag{1}$$

where the true label $y \in \mathcal{Y}$ indicates the semantic relation between the input sentence pair. $\mathcal{Y} = \{entailment, contradiction, neutral\}$ for NLI task and $\mathcal{Y} = \{Yes, No\}$ for PI task.

3.2 Relation of Relation Classification

Gururangan et al. (2018) has observed that relations can be helpful to reveal some implicit features or patterns for semantic understanding and matching. In order to properly and fully utilize relation information, we propose a **R**elation of **R**elation (R^2) classification task to guide models to understand sentence relation more precisely. Given two input sentence pairs $(\boldsymbol{s}_1^a, \boldsymbol{s}_1^b)$ and $(\boldsymbol{s}_2^a, \boldsymbol{s}_2^b)$, the goal is to learn a classifying function $\mathcal{F}$ with the ability to identify whether these two input pairs have the same semantic relation:

$$\mathcal{F}((\boldsymbol{s}_1^a, \boldsymbol{s}_1^b), (\boldsymbol{s}_2^a, \boldsymbol{s}_2^b)) = \begin{cases} 1, & \text{if } y_1 = y_2, \\ 0, & \text{if } y_1 \neq y_2, \end{cases} \tag{2}$$

where y_1 and y_2 stand for the semantic relations of two input sentence pairs, respectively.

In order to make full use of relation information and do better sentence semantic matching, the following important questions should be considered:

- Since relations are the predicting targets, how to make full use of relation information to improve model performance properly without leaking it?
- How to integrate R^2 task into matching task effectively for relation usage and performance improvement?

To this end, we propose R^2*-Net* to properly and fully utilize relation information, and tackle the above issues. Next, we will introduce the technical details of R^2*-Net*.

4 Relation of Relation Learning Network (R^2-Net)

The overall architecture of R^2*-Net* is shown in Figure 2(A). To better describe how R^2*-Net* tackles the above tasks and integrates R^2 task to enhance the model ability on sentence semantic matching, similar to Section 3, we also elaborate the technical details from two aspects: 1) Sentence Semantic Matching Part; 2) Relation of Relation Learning Part.

4.1 Sentence Semantic Matching Part

This part focuses on identifying the most suitable label for a given input sentence pair. Specifically, for an input sentence pair, we first utilize powerful BERT to generate sentence semantic representation globally. Meanwhile, we develop a CNN-based encoder to capture the keywords and phrase information from a local perspective. Thus, the input sentence pair can be encoded in a comprehensive manner. Based on the comprehensive representation, we leverage a multi-layer perceptron to predict the corresponding label.

Global Encoding. With the full usage of large corpus and multi-layer transformers, BERT (Devlin et al. 2018) has accomplished much progress in many NLP tasks. Thus, we select BERT to generate sentence semantic representations for the input. Moreover, inspired by ELMo (Peters et al. 2018), we also use the weighted sum of all the hidden states of words from different transformer layers as the final contextual representations of input words in sentences.

Specifically, we first split the input sentence pair $(\boldsymbol{s}^a, \boldsymbol{s}^b)$ into BPE tokens (Sennrich et al. 2015). Then, we concatenate two sentences to the required format, in which "[*SEP*]" is adopted to concatenate two sentences and "[*CLS*]" is added at the beginning and the end of the whole sequence. Then, we use multi-layer transformer blocks to obtain the representations of words and sentences in the input. Moreover, as illustrated in Figure 2(B), suppose there are L layers in the BERT. The contextual word representations in the input sentence pairs is then a pre-layer weighted sum of transformer block output, with the weights $\alpha_1, \alpha_2, ..., \alpha_L$.

$$\begin{aligned} \boldsymbol{h}_0^l, \boldsymbol{H}^l &= TransformerBlock(\boldsymbol{s}^a, \boldsymbol{s}^b), \\ \boldsymbol{H} &= \sum_{l=1}^{L} \alpha_l \boldsymbol{H}^l, \quad \boldsymbol{v}_g = \boldsymbol{h}_0^L, \end{aligned} \tag{3}$$

where $\boldsymbol{h}_0^l$ denotes the representation of first token "[*CLS*]" at the l^{th} layer, and $\boldsymbol{v}_g$ denotes the global semantic representation of the input. $\boldsymbol{H}^l$ represents the sequence features of the whole input. α_l is the weight of the l^{th} layer in BERT and will be learned during model training.

Local Encoding. The semantic relation within the sentence pair is not only connected with the important words, but also affected by the local information (e.g., phrase and local structure). Though Bert leverages multi-layer transformers to perceive important words to the sentence pair, it still has some weaknesses in modeling local information. To alleviate these shortcomings, we develop a CNN-based local encoder to extract the local information from the input.

Figure 2(C) illustrates the structure of this local encoder. The input of this encoder is the output features $\boldsymbol{H}$ from global encoding. We use convolution operations with different composite kernels (e.g., bigram and trigram) to process these features. Each operation with different kernels is capable of modeling patterns with different sizes (e.g., *new couple, tall person*). Thus, we can obtain robust and abstract local features of the input sentence pair. Next, we leverage average pooling and max pooling to enhance these local features and concatenate them before sending them to a nonlinear transformation. Suppose we have K different kernel sizes, this process can be formulated as follows:

$$\begin{aligned} \boldsymbol{H}^k &= CNN_k(\boldsymbol{H}), \quad k = 1, 2, ..., K, \\ \boldsymbol{h}_{max}^k &= max(\boldsymbol{H}^k), \boldsymbol{h}_{avg}^k = avg(\boldsymbol{H}^k), \\ \boldsymbol{h}_{concat} &= [\boldsymbol{h}_{max}^1; \boldsymbol{h}_{avg}^1; ...; \boldsymbol{h}_{max}^K; \boldsymbol{h}_{avg}^K], \\ \boldsymbol{v}_l &= ReLu(\boldsymbol{W}\boldsymbol{h}_{concat} + \boldsymbol{b}), \end{aligned} \tag{4}$$

where CNN_k denotes the convolution operation with the k^{th} kernel. $[\cdot;\cdot]$ is the concatenation operation. $\boldsymbol{v}_l$ represents the local semantic representation of the input. $\{\boldsymbol{W}, \boldsymbol{b}\}$ are trainable parameters. $ReLu(\cdot)$ is the activation function.

After getting the global representation $\boldsymbol{v}_g$ and local representation $\boldsymbol{v}_l$, we investigate the different fusion methods to integrate them together, including simple concatenation, weighed concatenation, as well as weighted sum. Finally, we obtain that simple concatenation is flexible and can achieve comparable performance without adding more training parameters. Thus, we employ the concatenation $\boldsymbol{v} = [\boldsymbol{v}_g; \boldsymbol{v}_l]$ as the final semantic representation of the input sentence pair.

Label Prediction. This component is adopted to predict the label of input sentence pair, which is an essential part of traditional sentence semantic matching methods. To be specific, the input of this component is the semantic representation $\boldsymbol{v}$. We leverage a two-layer MLP to make the final classification, which can be formulated as follows:

$$P(y|(\boldsymbol{s}^a, \boldsymbol{s}^b)) = MLP_1(\boldsymbol{v}). \tag{5}$$

4.2 Relation of Relation Learning Part

This part aims at properly and fully using relation information of input sentence pairs to enhance the model performance on sentence semantic matching. In order to achieve this goal, we employ two critical modules to analyze the *pairwise relation* and *triplet based relation* simultaneously. Next, we will describe each module in detail.

Relation of Relation Classification. Inspired by self-supervised learning methods in BERT, we intend R^2-*Net* to make full use of relation information among input sentence pairs in a similar way. Therefore, we introduce R^2 classification task into sentence semantic matching. Instead of just identifying the most suitable relation of input sentence pairs, we plan to obtain more knowledge about the input sentence pair by analyzing the *pairwise relation* between the semantic representations ($\boldsymbol{v}_1$ for pair $(\boldsymbol{s}_1^a, \boldsymbol{s}_1^b)$, and $\boldsymbol{v}_2$ for pair $(\boldsymbol{s}_2^a, \boldsymbol{s}_2^b)$). Since a learnable nonlinear transformation between representations and loss substantially improves the model performance (Chen et al. 2020), we first transfer $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ with a nonlinear transformation. Then, we leverage heuristic matching (Chen et al. 2017a) to model the similarity and difference between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Next, we send the result $\boldsymbol{u}$ to a MLP with one hidden layer for final classification. This process is formulated as follows:

$$\begin{aligned} \bar{\boldsymbol{v}}_1 &= ReLu(\boldsymbol{W}_r\boldsymbol{v}_1 + \boldsymbol{b}_r), \\ \bar{\boldsymbol{v}}_2 &= ReLu(\boldsymbol{W}_r\boldsymbol{v}_2 + \boldsymbol{b}_r), \\ \boldsymbol{u} &= [\bar{\boldsymbol{v}}_1; \bar{\boldsymbol{v}}_2; (\bar{\boldsymbol{v}}_1 \odot \bar{\boldsymbol{v}}_2); (\bar{\boldsymbol{v}}_1 - \bar{\boldsymbol{v}}_2)], \\ P(\hat{y}|(\boldsymbol{s}_1^a, \boldsymbol{s}_1^b), (\boldsymbol{s}_2^a, \boldsymbol{s}_2^b)) &= MLP_2(\boldsymbol{u}), \end{aligned} \tag{6}$$

where concatenation can retain all the information (Zhang et al. 2017). The element-wise product is a certain measure of "similarity" of two sentences (Mou et al. 2016b). Their difference can capture the degree of distributional inclusion in each dimension (Weeds et al. 2014). $\hat{y} \in \{1, 0\}$ indicates whether two input sentence pairs have same relation.

Triplet Distance Calculation. Apart from leveraging R^2 classification task to learn pairwise relation information, we also intend to learn intra-class and inter-class information from the *triplet based relation*. Thus, we also introduce a triplet loss (Schroff et al. 2015) into R^2-*Net*. As a fundamental similarity function, triplet loss is widely applied in information retrieval area (Liu 2009), and is able to reduce the

Table 1: Performance (accuracy) of models on different NLI dataset.

Model	Full test	Hard test	SICK test
(1) CENN (Zhang et al. 2017)	82.1%	60.4%	81.8%
(2) CAFE (Tay, Tuan, and Hui 2017)	85.9%	66.1%	86.1%
(3) Gumbel TreeLSTM (Choi, Yoo, and Lee 2018)	86.0%	66.7%	85.8%
(4) Distance-based SAN (Im and Cho 2017)	86.3%	67.4%	86.7%
(5) DRCN (Kim et al. 2018)	86.5%	68.3%	87.4%
(6) DRr-Net (Zhang et al. 2019)	87.5%	71.2%	87.8%
(7) Dynamic Self-Attention (Yoon, Lee, and Lee 2018)	87.4%	71.5%	87.7%
(8) Bert-base (Devlin et al. 2018)	90.3%	80.8%	88.5%
(9) *R*^{2}*-Net*	**91.1%**	**81.0%**	**89.2%**

Table 2: Experimental Results (accuracy) on SciTail dataset.

Model	SciTail test
(1) CAFE (2017)	83.3%
(2) ConSeqNet (2018b)	85.2%
(3) BiLSTM Max-Out (2018)	85.4%
(4) HBMP (Talman et al.2018)	86.0%
(5) DRr-Net (2019)	87.4%
(6) Transformer LM (2018)	88.3%
(7) Bert-base (2018)	92.0%
(8) *R*^{2}*-Net*	**92.9%**

distance of input pairs with the same relation and increase the distance of these with different relations. Therefore, we first calculate the corresponding distances in this module. To be specific, the inputs of this component are three semantic representations: $\boldsymbol{v}_a$ for anchor pair $(\boldsymbol{s}_a^a, \boldsymbol{s}_a^b)$, $\boldsymbol{v}_p$ for positive pair $(\boldsymbol{s}_p^a, \boldsymbol{s}_p^b)$, $\boldsymbol{v}_n$ for negative pair $(\boldsymbol{s}_n^a, \boldsymbol{s}_n^b)$. In order to obtain better results, we first transform them into a common space with a full connection layer (Chen et al. 2020). Then, we calculate the distance between anchor and positive pairs, and the distance between anchor and negative pairs, respectively. This process is formulated as follows.

$$\begin{aligned}
&\bar{\boldsymbol{v}}_a = ReLu(\boldsymbol{W}_d \boldsymbol{v}_a + \boldsymbol{b}_d),\\
&\bar{\boldsymbol{v}}_p = ReLu(\boldsymbol{W}_d \boldsymbol{v}_p + \boldsymbol{b}_d),\\
&\bar{\boldsymbol{v}}_n = ReLu(\boldsymbol{W}_d \boldsymbol{v}_n + \boldsymbol{b}_d),\\
&d_{ap} = Dist(\bar{\boldsymbol{v}}_a, \bar{\boldsymbol{v}}_p), \quad d_{an} = Dist(\bar{\boldsymbol{v}}_a, \bar{\boldsymbol{v}}_n),
\end{aligned} \tag{7}$$

where $\{\boldsymbol{W}_d, \boldsymbol{b}_d\}$ are trainable parameters. $Dist(\cdot)$ is the distance calculation function.

5 Experiments

In this section, the details about model implementation are firstly presented. Then, five benchmark datasets on which the model is evaluated are introduced. Next, a detailed analysis about the model and experimental results is made.

5.1 Training Details

Loss Function. As is mentioned in Section 3, both sentence semantic matching and R^2 task can be treated as classification tasks. Thus, we employ *Cross-Entropy* as the loss for each input as follows:

$$\begin{aligned}
L_s &= -\boldsymbol{y}_i \log P(y_i | (\boldsymbol{s}_i^a, \boldsymbol{s}_i^b)),\\
L_{R^2} &= -\hat{\boldsymbol{y}}_i \log P(\hat{y} | ((\boldsymbol{s}_1^a, \boldsymbol{s}_1^b), (\boldsymbol{s}_2^a, \boldsymbol{s}_2^b))_i),
\end{aligned} \tag{8}$$

where $\boldsymbol{y}_i$ is the one-hot vector for the true label of the i^{th} instance. $\hat{\boldsymbol{y}}_i$ is the one-hot vector for the true relation of relations of the i^{th} instance pair.

Moreover, in order to learn more from relations and achieve better performance, we also introduce the triplet loss to force *R*^{2}*-Net* to better analyze the intra-class and inter-class information among sentence pairs with same or different relations:

$$L_d = max((d_{ap} - d_{an} + \alpha)_i, 0), \tag{9}$$

where α is the margin. $(\cdot)_i$ denotes the i^{th} triplet pair.

Since these three loss functions require different number of inputs, we modify the input of *R*^{2}*-Net* to have three input sentence pairs (i.e., anchor pair, positive pair, and negative pair), as shown in Figure 2(A). Then, we calculate L_s^1, L_s^2, L_s^3 for label prediction loss of each input pair, randomly sample two groups from the input to calculate $L_{R^2}^1, L_{R^2}^2$ for R^2 task loss, and use three input pairs to calculate L_d for triplet loss. Finally, we treat the weighed sum of these losses with a hyper-parameter β as the loss function for entire model as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N}(\beta\frac{L_s^1 + L_s^2 + L_s^3}{3} + (1-\beta)(\frac{L_{R^2}^1 + L_{R^2}^2}{2} + L_d)). \tag{10}$$

Model Implementation. We have tuned the hyper-parameters on validation set for best performance, and have used early-stop to select the best model. Since *R*^{2}*-Net* has different hyper-parameter settings on different datasets, we list some common hyper-parameters as follows.

We apply the BERT-base with 12 layers, hidden size 768, and 12 heads. The kernel sizes of CNN in local encoding unit are $d_k = 1, 2, 3$. The hidden state size of MLP in *R*^{2}*-Net* is $d_m = 300$. The distance we use in distance calculation component is *Euclidean Distance*. The margin α in the triplet loss is $\alpha = 0.2$. For the pre-trained BERT, we set the learning rate 10^{-5} and use AdamW to fine-tune the parameters. For the rest of parameters, we set the initial learning rate to be 10^{-3} and decrease its value as the model training. An Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is adopted to optimize these parameters. The entire model is implemented with PyTorch and Transformers[1], and is trained on two Nvidia Tesla V100-SXM2-32GB GPUs.

[1] https://github.com/huggingface/transformers

Table 3: Experimental Results (accuracy) on Quora and MSRP datasets.

Model	Quora test	MSRP test
(1) CENN (Zhang et al. 2017)	80.7%	76.4%
(2) L.D.C (Wang, Mi, and Ittycheriah 2016)	85.6%	78.4%
(3) REL-TK (Filice et al. 2015)	-	79.1%
(4) BiMPM (Wang, Hamza, and Florian 2017)	88.2%	-
(5) pt-DecAttachar.c (Tomar et al. 2017)	88.4%	-
(6) DIIN (Gong, Luo, and Zhang 2017)	89.1%	-
(7) DRr-Net (Zhang et al. 2019)	89.8%	82.9%
(8) DRCN (Kim et al. 2018)	90.2%	82.5%
(9) BERT-base (Devlin et al. 2018)	91.0%	84.2%
(10) R^2-*Net*	**91.6%**	**84.3%**

5.2 Data Description

In this section, we give a brief introduction of the datasets on which we evaluate all models. They are as follow:

- **SNLI:** SNLI dataset (Bowman et al. 2015) contains $570,152$ human annotated sentence pairs. The premise sentences are drawn from the captions of Flickr30k corpus (Young et al. 2014), and the hypothesis sentences are manually composed. Despite the original test set, we also select the challenging hard subset (Gururangan et al. 2018) to evaluate the models.
- **SICK:** SICK dataset (Marelli et al. 2014) contains $10,000$ English sentence pairs, generated from 8K ImageFlickr dataset (Hodosh et al. 2013) and STS MSR-video description dataset[2]. Each sentence pair is generated from randomly selected subsets of the above sources and manually labeled with the label set as SNLI did.
- **SciTail**: SciTail dataset (Khot et al. 2018) is created from multiple-choice science exams and web sentences. It has $27,026$ examples with $10,101$ *Entailment* examples and $16,925$ *Neutral* examples.
- **Quora**: Quora dataset (Iyer et al. 2017) contains over $400,000$ potential question duplicate pairs, which are drawn from Quora website[3]. This dataset has balanced positive and negative labels, indicating whether the line truly contains a duplicate pair.
- **MSRP**: MSRP dataset (Dolan and Brockett 2005) consists of $5,801$ sentence pairs with a binary label. The sentences are distilled from a database of $13,127,938$ sentence pairs, extracted from $9,516,684$ sentences in $32,408$ news clusters from the web.

5.3 Experimental Results

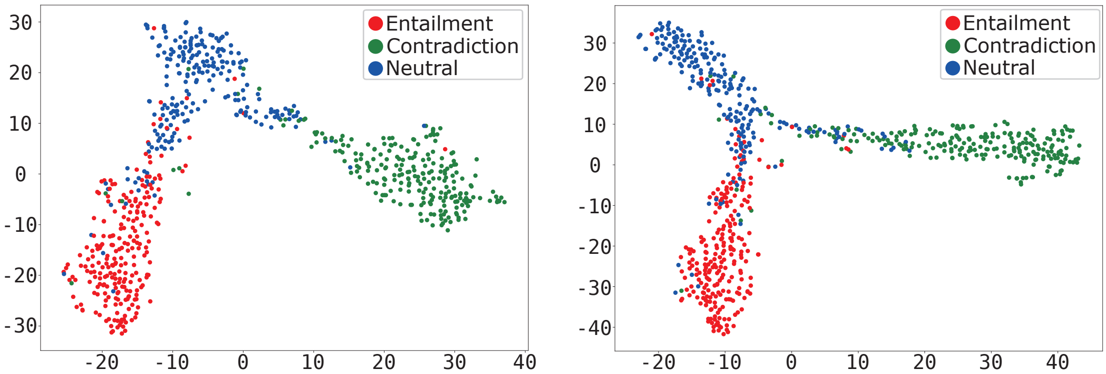
In this section, we will give a detailed analysis about models and experimental results. We have to note that we use *accuracy* on different test sets to evaluate the model performance.

Performance on NLI task. We compared our proposed R^2-*Net* to several published state-of-the-art baselines on different NLI datasets. All results are summarized in Table 1 and Table 2. Several observations are presented as follows.

- It is clear that R^2-*Net* achieves highly comparable performance over all the datasets: SNLI, SICK, and SciTail. Specifically, R^2-*Net* first fully uses BERT and CNN-based encoder to get a comprehensive understanding of sentence semantics from global and local perspectives. This is one of the reasons that R^2-*Net* outperforms other BERT-free baselines by a large margin. Another important reason is that R^2-*Net* employs R^2 task and triplet loss to make full use of relation information. Along this line, R^2-*Net* is capable of obtaining intra-class and inter-class knowledge among sentence pairs with the same or different relations. Thus, it can achieve better performance than all baselines, including the BERT-base model.
- R^2-*Net* has more stable performance on the challenging NLI hard test, in which the pairs with obvious identical words are removed (Gururangan et al. 2018). Despite the obvious indicators, these still have implicit patterns for relations among sentence pairs. By considering R^2 task and triplet loss, R^2-*Net* has the ability to fully use relation information and obtain the implicit information, which leads to a better performance.
- BERT-base model (Devlin et al. 2018) outperforms other BERT-free baselines by a large margin. The main reasons can be grouped into two parts. First, BERT takes advantages of multi-layer transformers to learn sentence patterns and sentence semantics on a large corpus. Second, BERT adopts two self-supervised learning tasks (i.e., MLM and NSP) to better analyze the important words within a sentence and semantic connection between sentences. However, BERT still focuses on the input sequence, underestimating the rich semantic information that relations imply. Therefore, its performance is not as good as that R^2-*Net* achieves.
- Among BERT-free baselines, DRr-Net (Zhang et al. 2019) and dynamic self-attention (Yoon, Lee, and Lee 2018) achieve impressive performances. First, their performance proves that multi-layer structure and CNN have better ability to model sentence semantics from global and local perspectives. Then, they all develop a dynamic attention mechanism to improve self-attention mechanism. However, their encoding capability of extracting features or generating semantic representations is still not comparable with BERT. This observation inspires us that using powerful BERT as a basic encoder will be a better choice.

[2] https://www.cs.york.ac.uk/semeval-2012/

[3] https://www.quora.com/



(A)T-SNE of results from our proposed model

(B)T-SNE of results from BERT-base model

Figure 3: Visualization of representation $\boldsymbol{v}$ from R^2-*Net* and BERT-base models.

Table 4: Ablation performance (accuracy) of R^2-*Net*.

Model	SNLI test	SciTail test
(1) Bert-base	90.3%	92.0%
(2) R^2-*Net* (w/o local encoder)	90.7%	92.6%
(3) R^2-*Net* (w/o R^2 task learning)	90.5%	92.3%
(4) R^2-*Net* (w/o triplet loss)	90.9%	92.6%
(5) R^2-*Net*	**91.1%**	**92.9%**

Performance on PI task. Apart from NLI task, we also select PI task to evaluate the model performance. PI task concerns whether two sentences express the same meaning and has broad applications in question answering communities[4] [5]. Table 3 reports the performance of models on different datasets. We also list the observations as follows:

- R^2-*Net* still achieves highly competitive performance over other baselines. The results demonstrate that R^2 task and triplet loss is effective in helping our proposed model to learn more about relations and improve the model performance, even if the number of relations is small.
- Almost all models have a better performance on Quora dataset than MSRP dataset. One possible reason is that Quora dataset has more data than MSRP dataset (over 400k sentences pairs v.s. 5,801 sentence pairs). In addition to the data size, inter-sentence interaction is probably another reason. Lan et al. (2018) observes that Quora dataset contains many sentence pairs with less complicated interactions (many identical words in sentence pairs). Meanwhile, R^2-*Net* also achieves better improvement on Quora dataset, indicating that more data or better label utilization method is needed for further performance improvement on MSRP dataset.

Ablation Performance. The overall experiments have proved the superiority of R^2-*Net*. However, which part plays a more important role in performance improvement is still unclear. Therefore, we perform an ablation study to verify the effectiveness of each part, including *CNN-based local encoder*, R^2 *task classification*, and *triplet loss*. The results are illustrated in Table 4. Note that we select BERT-base as the baseline to compare the importance of each part. According to the results, we can observe varying degrees of model performance decline. Among all of them, R^2 task has the biggest impact, and triple loss has a relatively small impact on the model performance. These observations prove that R^2 task is more important for relation information utilization.

Case Study. To provide some intuitionistic examples for explaining why our model gains a better performance than other baselines, we sample 700 sentence pairs from SNLI dataset and send them to R^2-*Net* and BERT-base models to generate the semantic representation $\boldsymbol{v}$. Then, we leverage t-sne (Maaten and Hinton 2008) to visualize these representations with the same parameter settings. Figure 3(A)-(B) report the results of R^2-*Net* and BERT-base models, respectively. By comparing two figures, we can obtain that the representations generated by R^2-*Net* have closer inter-class distances. Moreover, the representations have more obvious distinctions between different classes. These observations not only explain why our proposed R^2-*Net* achieves impressive performance, but also demonstrates that proper usage of relation information is able to guide models to analyze sentence semantics more comprehensively and precisely, which is in favor of tackling sentence semantic matching.

6 Conclusion

In this paper, we presented a simple but effective method named R^2-*Net* for sentence semantic matching. This method not only uses powerful BERT and CNN to encode sentences from global and local perspectives, but also makes full use of relation information for better performance enhancement. Specifically, we design a R^2 classification task to help R^2-*Net* for learning the implicit common knowledge from the pairwise relation learning processing. Moreover, a triplet loss is employed to constrain R^2-*Net* for better triplet based relation learning and intra-class and inter-class information analyzing. Extensive experiments on NLI and PI tasks demonstrate the superiority of R^2-*Net*. In the future, we plan to combine the advantages of label embedding method for better sentence semantic comprehension.

[4] https://www.quora.com/

[5] https://www.zhihu.com/

7 Acknowledgments

This research was partially supported by grants from the National Natural Science Foundation of China (Grant No. 62006066, 61725203, 61972125, 61732008, and U19A2079), the Major Program of the National Natural Science Foundation of China (91846201), the Fundamental Research Funds for the Central Universities, HFUT. The author wants to thank Cui Gongrongxiu for constructive criticism on the earlier draft.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Bowman, S. R.; Gauthier, J.; Rastogi, A.; Gupta, R.; Manning, C. D.; and Potts, C. 2016. A Fast Unified Model for Parsing and Sentence Understanding. In *ACL*.

Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017a. Enhanced LSTM for Natural Language Inference. In *ACL*, 1657–1668.

Chen, Q.; Zhu, X.-D.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017b. Recurrent Neural Network-Based Sentence Encoder with Gated Attention for Natural Language Inference. In *RepEval@EMNLP*, 36–40.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *ArXiv* abs/2002.05709.

Cho, K.; Courville, A. C.; and Bengio, Y. 2015. Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks. *IEEE Trans. Multimedia* 17: 1875–1886.

Choi, J.; Yoo, K. M.; and Lee, S.-g. 2018. Learning to compose task-specific tree structures. In *AAAI*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555.

Clark, P.; Etzioni, O.; Khot, T.; Sabharwal, A.; Tafjord, O.; Turney, P. D.; and Khashabi, D. 2016. Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions. In *AAAI*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dolan, W. B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP*.

Du, C.; Chen, Z.; Feng, F.; Zhu, L.; Gan, T.; and Nie, L. 2019. Explicit interaction model towards text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6359–6366.

Filice, S.; Da San Martino, G.; and Moschitti, A. 2015. Structural representations for learning relations between pairs of texts. In *ACL*, 1003–1013.

Gong, Y.; Luo, H.; and Zhang, J. 2017. Natural Language Inference over Interaction Space. *CoRR* abs/1709.04348.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S. R.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *NAACL-HLT*, 107–112.

He, H.; Gimpel, K.; and Lin, J. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*, 1576–1586.

Hodosh, M.; Young, P.; and Hockenmaier, J. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47: 853–899.

Im, J.; and Cho, S. 2017. Distance-based Self-Attention Network for Natural Language Inference. *CoRR* abs/1712.02047.

Iyer, S.; Dandekar, N.; and Csernai, K. 2017. First quora dataset release: Question pairs.

Khot, T.; Sabharwal, A.; and Clark, P. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Kim, S.; Hong, J.-H.; Kang, I.; and Kwak, N. 2018. Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information. *CoRR* abs/1805.11360.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Kun, Z.; Guangyi, L.; Le, W.; Enhong, C.; Qi, L.; and Han, W. 2018. Image-Enhanced Multi-Level Sentence Representation Net for Natural Language Inference. In *IEEE ICDM*.

Lan, W.; and Xu, W. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *COLING*, 3890–3902.

Liu, Q.; Huang, Z.; Huang, Z.; Liu, C.; Chen, E.; Su, Y.; and Hu, G. 2018. Finding Similar Exercises in Online Education Systems. In *SIGKDD*, 1821–1830. ACM.

Liu, T.-Y. 2009. Learning to rank for information retrieval. In *Found. Trends Inf. Retr.*, volume 3, 225–331.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504* .

Liu, Y.; Sun, C.; Lin, L.; and Wang, X. 2016. Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention. *CoRR* abs/1605.09090.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Marelli, M.; Bentivogli, L.; Baroni, M.; Bernardi, R.; Menini, S.; and Zamparelli, R. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval*, 1–8.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *EMNLP*.

Mou, L.; Men, R.; Li, G.; Xu, Y.; Zhang, L.; Yan, R.; and Jin, Z. 2016a. Natural Language Inference by Tree-Based Convolution and Heuristic Matching .

Mou, L.; Men, R.; Li, G.; Xu, Y.; Zhang, L.; Yan, R.; and Jin, Z. 2016b. Natural Language Inference by Tree-Based Convolution and Heuristic Matching. In *ACL*, 130–136.

Pappas, N.; and Henderson, J. 2019. GILE: A Generalized Input-Label Embedding for Text Classification. *Transactions of the Association for Computational Linguistics* 7: 139–155.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*, 2249–2255.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *ArXiv* abs/1802.05365.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving Language Understanding by Generative Pre-Training.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 815–823.

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* .

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI*, volume 16.

Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; and Zhang, C. 2017. DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding. *CoRR* abs/1709.04696.

Talman, A.; Yli-Jyrä, A.; and Tiedemann, J. 2018. Natural language inference with hierarchical bilstm max pooling architecture. *arXiv preprint arXiv:1808.08762* .

Tay, Y.; Tuan, L. A.; and Hui, S. C. 2017. A Compare-Propagate Architecture with Alignment Factorization for Natural Language Inference. *CoRR* abs/1801.00102.

Tomar, G. S.; Duque, T.; Täckström, O.; Uszkoreit, J.; and Das, D. 2017. Neural Paraphrase Identification of Questions with Noisy Pretraining. In *SWCN@EMNLP*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; and Carin, L. 2018a. Joint Embedding of Words and Labels for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2321–2331.

Wang, X.; Kapanipathi, P.; Musa, R.; Yu, M.; Talamadupula, K.; Abdelaziz, I.; Chang, M.; Fokoue, A.; Makni, B.; Mattei, N.; and Witbrock, M. J. 2018b. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. In *AAAI*.

Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. *CoRR* abs/1702.03814.

Wang, Z.; Mi, H.; and Ittycheriah, A. 2016. Sentence Similarity Learning by Lexical Decomposition and Composition. In *COLING*.

Weeds, J.; Clarke, D.; Reffin, J.; Weir, D.; and Keller, B. 2014. Learning to distinguish hypernyms and co-hyponyms. In *COLING*, 2249–2259.

Yoon, D.; Lee, D.; and Lee, S. 2018. Dynamic self-attention: Computing attention over words dynamically for sentence embedding. *arXiv preprint arXiv:1808.07383* .

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL* 2: 67–78.

Zhang, H.; Xiao, L.; Chen, W.; Wang, Y.; and Jin, Y. 2018. Multi-Task Label Embedding for Text Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4545–4553.

Zhang, K.; Chen, E.; Liu, Q.; Liu, C.; and Lv, G. 2017. A Context-Enriched Neural Network Method for Recognizing Lexical Entailment. In *AAAI*, 3127–3133.

Zhang, K.; Lv, G.; Wang, L.; Wu, L.; Chen, E.; Wu, F.; and Xie, X. 2019. Drr-net: Dynamic re-read network for sentence semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7442–7449.

Zheng, Y.; Mao, J.; Liu, Y.; Ye, Z.; Zhang, M.; and Ma, S. 2019. Human behavior inspired machine reading comprehension. In *SIGIR*, 425–434.