International Journal of Innovative Computing, Information and Control
Volume **15**, Number **5**, October **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **1685–1700**

SEMANTIC SIMILARITY MODELING BASED ON MULTI-GRANULARITY INTERACTION MATCHING

Xu Li*, Chunlong Yao, Qinyang Zhang and Guoqi Zhang

School of Information Science and Engineering
Dalian Polytechnic University
No. 1, Qinggongyuan, Ganjingzi District, Dalian 116304, P. R. China
*Corresponding author: lixu102@aliyun.com

Received February 2019; revised June 2019

Abstract. *Determining whether two sentences are semantically equivalent is complicated by the ambiguity and variability of natural language expression. The most approaches have used classifiers employing hand engineered features derived from complex natural language processing pipelines to automatically recognize equivalence relations; thus, the performances of the models heavily rely on the features designing. To avoid specific assumptions about the underlying language, we propose a recurrent neural network model for semantic similarity. Interaction features and text representations on multiple levels of granularity are automatically learned using a conditional bidirectional long short-term memory encoder. We extend this model with a soft-alignment attention mechanism that encourages fine-grained reasoning over equivalence or contradiction of pairs of words and phrases. The sentence-pair encoding is input to an output layer to determine the classification. The effectiveness of our model is demonstrated using two tasks: paraphrase identification and semantic relatedness measurement. The results on MRPC and SICK datasets show that our model leads to significant quality improvement on tasks, exceeding the previous state-of-the-art without using any hand-crafted features.*
Keywords: Semantic similarity, Deep learning, Recurrent neural network, Attention mechanism

1. **Introduction.** Semantic similarity or paraphrase identification involves predicting whether two sentences are semantically equivalent or not. This task is important since many natural language processing tasks, such as QA (Question Answering), MT (Machine Translation), text summarization or information retrieval, rely on it explicitly or implicitly and could benefit from more accurate semantic equivalence identification systems. In the question answering and dialogue system, the user's input is very casual and colloquial. Paraphrase identification technology can match the user's questions to the standard questions and improve the recall of answer extraction. In the machine translation, a successful sentence matching algorithm can help the MT system to evaluate the semantic similarity between the machine generated sample and the reference, and then accurately score for the generated translation. In the automatic summarization, efficient recognition of sentences with the same or similar meaning can better perform sentence clustering, abstract sentence selection, and generate more accurate and concise abstracts. In addition, an accurate semantic equivalence identification system is also helpful to detect plagiarism between texts, which can effectively resist academic misconduct and promote the construction of scientific integrity.

DOI: 10.24507/ijicic.15.05.1685

The challenges of this task lie in rephrasing of concepts, understanding negation, and handling syntactic ambiguity. Conventional semantic similarity approaches required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data into a suitable internal representation or feature vector, could binary classify patterns in the input. However, these hand engineered features were always designed for specific tasks and struggled with out-of-domain data, which were difficult to employ to other natural language processing tasks. Inspired by recent successes of deep neural networks in fields like image recognition, speech recognition and natural language processing, we adopt a deep learning approach to recognize semantic relations between two sentences in this paper. The key aspect of deep learning is that features are not designed by human engineers which are learned from data using a general purpose learning procedure. Deep learning approaches are representation learning with multiple levels of representation, obtained by composing simple but non-linear modules that each transforms the representation at one level into a representation at a higher, slightly more abstract level. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations.

The existing deep semantic similarity model, such as ARC-I [1] or He et al. [2], usually focuses on semantic representation of text. It first finds the representation of each sentence, and then compares the matching degree between the semantic representations of the two sentences. Although the approach is simple in semantic expression and fast in calculation, it suffers from a drawback: it defers the interaction between two sentences to until their individual representation matrices, therefore runs at the risk of losing details important for the semantic matching task in representing the sentences. In other words, in the forward phase (prediction), the representation of each sentence is formed without knowledge of each other. This cannot be adequately circumvented in backward phase (learning), when the model learns to extract semantic information for matching on a population level.

In view of the drawback of above approach, we built directly on the interaction space between two sentences. In fact, humans always focus on the interaction matching of texts when they compare the meaning of two sentences. They usually first find out whether there are matching keywords in the two sentences, and then observe the relative positions of the keywords in the sentences, and finally integrate the meaning of the whole sentence to score the matching degree. Semantic similarity modeling based on interaction space has the desirable property of letting two sentences meet before their own high-level representations matrix, while still retaining the space for the individual development of abstraction of each sentence. The approach is built on the interaction between two sentences, which offers not only the extraction and fusion of matching patterns between them but also the inductive bias for the individual development of internal abstraction on each sentence. Hence, semantic similarity modeling based on interaction learning is powerful for matching linguistic objects with rich structures.

Another motivation for our work lies in the key observation that the identification of an equivalence relationship between two sentences requires reasoning at multiple levels of granularity such as word, phrase, and sentence.

(A1) Detroit manufacturers have raised vehicle prices by ten percent.
(A2) GM, Ford and Chrysler have raised car prices by five percent.
(B1) Dancing was the only physical activity associated with a lower risk of dementia.
(B2) With the exception of dancing physical activity did not decrease the risk.
(C1) Mary gave birth to a son in 2012.
(C2) He is 7 years old and his mother is Mary.

Example A1/A2 shows that semantic similarity requires comparison at the word level. A1 cannot be a paraphrase of A2 because the numbers "ten" and "five" are contradictory. Paraphrase identification for B1/B2 can succeed since "only" and "with the exception of" are connected via deeper semantics and "a lower risk" and "decrease the risk" are matched phrase expressions to the same meaning. Semantic similarity for C1/C2 can succeed at the sentence level since C1/C2 express the same meaning using very different means. Most work on semantic similarity has focused on only one level of granularity: either on low-level features or on the entire sentence level, for example, Madnani et al. [3] and ARC-I. An exception is the Bi-CNN-MI [4]. It learns and computes representations at multiple levels of granularity using CNN (Convolutional Neural Network). However, CNN cannot capture the long-distance dependence of sentences, which can be effectively solved by using RNN (Recurrent Neural Network). The letters and words in natural language appear one by one, and the relative order between words is very important. Therefore, we can regard natural language sentence as sequence data with time series dependence and recognize semantic relations using recurrent neural network architecture.

Our contributions are threefold. (i) We focus on text interaction matching pattern and present a neural model based on Bi-LSTM (Bidirectional Long Short-Term Memory) that reads two sentences in one interaction space go to determine equivalence, as opposed to mapping each sentence independently into a semantic space. (ii) We propose a soft-alignment attention mechanism because the contribution of memory cell at each time step of the recurrent neural network to classifier is different. The proposed model employs the attention mechanism, which assigns different weights to the hidden layer outputs of at different time steps, in the pooling phase and pays attention to semantically related pairs of text. The attention mechanism encourages fine-grained reasoning at the word, phrase and sentence level and achieves satisfactory learning performances. (iii) The accuracy of the proposed model is 0.788 and the $F1$ score reaches 0.848 on the public MRPC (Microsoft Paraphrase Corpus), which outperforms the classifiers with human engineered features and the previous best neural model. The Pearson correlation coefficient of semantic relatedness measurement between two sentences is 0.899 on the SICK (Sentences Involving Compositional Knowledge) provided in SemEval-2014 task 1, which sets a new state-of-the-art for predicting the degree of relatedness between two sentences on SICK. The model is an end-to-end differentiable system without using any hand-crafted features and it has a good generalization performance.

Section 2 discusses related work. Section 3 introduces the recurrent neural network model for semantic similarity. Section 4 describes the experiment settings and analyzes results. Conclusion and future work are presented in Section 5.

2. **Related Work.** Most previous work on modeling sentence similarity has focused on feature engineering. Finch et al. [5] proposed an approach based on bag-of-words model, it used WER (Word Error Rate), PER (Position-independent word Error Rate), BLEU score, NIST score and POS (Part-of-Speech) enhanced RER as a feature of classification using SVM (Support Vector Machine). The sentences were first tokenized and then POS tagged, while stemming was performed only on nouns and verbs. Moreover, they used semantic similarity distance measure computed based on WordNet lexical relationship measures. Qiu et al. [6] presented a framework of two-phase for paraphrase identification. The first phase identified the common content information of the pair of sentences using similarity detection. Then the information was paired using a pairing module. This common information content was called information nuggets, it was provided in a tuple of predicated argument form. Using a simple matching technique, the predicate arguments were compared. This approach is different from other approaches because it

is focusing on dissimilarities between the pairs of sentences. Hassan [7] produced an S-SA (Salient Semantic Analysis) model for measuring the semantic relatedness of words. They used salient encyclopedic features taken from encyclopedic knowledge to construct a semantic profile for these words. This approach is built on the idea that the meaning of a word can be represented in a salient concept found in its immediate context. Eyecioglu and Keller [8] group was one of the teams that participated in SemEval-2015 Task1 Workshop. In their approach for paraphrase identification they used an SVM along with simple lexical overlap features of words and characters based on n-gram. Their results showed that they achieved the highest performance in paraphrase identification task. This indicates the important role lexical overlap features could play in enhancing the results of paraphrase identification. Several approaches used system combination or multi-task learning. Xu et al. [9] developed a feature-rich multi-instance learning model that jointly learns paraphrase relations between word and sentence pairs. While these conventional approaches used hand engineering features derived from complex natural language processing pipelines, in practice their performances have been only slightly better than bag-of-word pair classifiers using only lexical similarity because of the ambiguity and variability of natural language expression.

Recent work has moved away from hand-crafted features and towards modeling with distributed representations and neural network architectures. Lu and Li [10] proposed a deep neural network to match short texts, where interactions between components within the two objects were considered. These interactions were obtained via LDA (Latent Dirichlet Allocation). A two-dimensional interaction space was formed, those local decisions would be sent to the corresponding neurons in upper layers to get rounds of fusion, and finally logistic regression in the output layer produced the final matching score. Drawbacks of this approach are that LDA parameters are not optimized for the paraphrase task and that the interactions are formed on the level of single words only. Gao et al. [11] presented a DSSM (Deep Semantic Similarity Model) which mapped source-target document pairs to feature vectors in latent space in such a way that the distance between source documents and their corresponding interesting targets in that space was minimized. The model is a document-level model and it is not multi-granular. Hu et al. [1] used convolutional neural networks that combined hierarchical sentence modeling with layer-by-layer composition and pooling. While they performed comparisons directly over entire sentence representations, local comparisons were not considered. Yin and Schutze [4] presented a deep learning architecture Bi-CNN-MI for paraphrase identification, which we compare to in our experiments. They learned multi-granular sentence representations using convolutional neural network and modeled interaction features at each level. Their best results rely on an unsupervised pre-training step, which we do not need to match their performance. Tien et al. [12] proposed an M-MaxLSTM-CNN model for evaluating sentence relation. Representing each word by multiple word embeddings, the MaxLSTM-CNN encoder generated a novel sentence embedding. They then learned the relations between the sentence embeddings via multi-level comparison. The accuracy and $F1$ score of the model are significantly lower than our model. Lan and Xu [13] analyzed several neural network designs for sentence-pair modeling and compared their performance extensively across eight datasets. Their study has been shown that (i) encoding contextual information by LSTM and inter-sentence interactions are critical, (ii) the enhanced sequential inference model is the best so far for large datasets, while the pairwise word interaction model achieves the best performance when less data is available. The research in this paper also proves the above findings. For the paraphrase identification task, the multi-perspective sentence similarity modeling proposed by He et al. [2] performed best

in the previous neural network models. They first modeled each sentence using a convolutional neural network that extracted features at multiple levels of granularity and used multiple types of pooling. They then compared the sentence representations at several granularities using multiple similarity metrics. The drawback of this approach has been discussed in Section 1. Additionally, they used multiple kinds of embeddings to represent each sentence in their experiments, both on words and part-of-speech tags. Some linguistic features such as part-of-speech and named entity annotation are usually input into the model for better performance. However, the extraction of the above linguistic information still relies on considerable hand feature engineering. In order to avoid hand design and feature extraction, our model does not use any other language features except for the public pre-training word embeddings, which can be downloaded directly from the public web pages and used. Compared with the multi-perspective sentence similarity modeling, the accuracy and the $F1$ score of our model are improved, and the best performance is achieved in all neural network models.

3. **Sentence-Pair Modeling Based on Recurrent Neural Network.** The semantic similarity is defined as follows.

Input: (s_1, s_2), $s_1 \in S$, $s_2 \in S$, where s_1 and s_2 are two sentences, and S is the set of sentences.

Output: $y \in Y$, $Y = \{0, 1\}$, where 0 denotes that the meanings of two sentences are different and 1 indicates that two sentences are semantically equivalent.

Training set: $D = \left\{ \left(s_1^{(1)}, s_2^{(1)}, y^{(1)} \right), \ldots, \left(s_1^{(i)}, s_2^{(i)}, y^{(i)} \right), \ldots, \left(s_1^{(n)}, s_2^{(n)}, y^{(n)} \right) \right\}$, $i = \{1, 2, \ldots, n\}$, where $s_1^{(i)} \in S$, $s_2^{(i)} \in S$, $y^{(i)} \in Y$.

The goal of the semantic similarity model is to automatically learn the matching function $f : S \times S \to Y$ in the training sets. For any input (s_1', s_2') ($s_1' \in S$ and $s_2' \in S$) on the test set, the model can predict the category label y' ($y' \in Y$) describing whether the two sentences express the same meaning.

Neural networks are good at calculating precise mapping functions from input to output for sufficient training samples. However, specific problems can only be effectively solved by appropriate neural network architectures, inputs and outputs. In this paper, we present the recurrent neural network combined with a soft-alignment attention mechanism to train the semantic similarity model. The architecture of our model is shown in Figure 1.

The architecture consists of four layers: input layer, hidden layer, attention layer and output layer. The input layer converts the words in the sentence into corresponding vector representations and passes them to the hidden layer. The hidden layer uses Bi-LSTMs to conditionally encode two candidate paraphrases to obtain textual semantic representations containing historical and future context information at each time step. The purpose of the attention layer is to learn weights for the hidden layer outputs at each time step, so that the learning system focuses on the useful information in the input data that is significantly related to the current output. The output layer performs binary classification based on the normalized probabilities generated by the softmax activation function, and outputs the predicted results.

3.1. **Combined pre-training word embedding.** Word embedding refers to the mapping words from the vocabulary to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension. We use combination of GloVe (Global Vectors) [14] and PARAGRAM [15] word embeddings to represent each sentence. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and

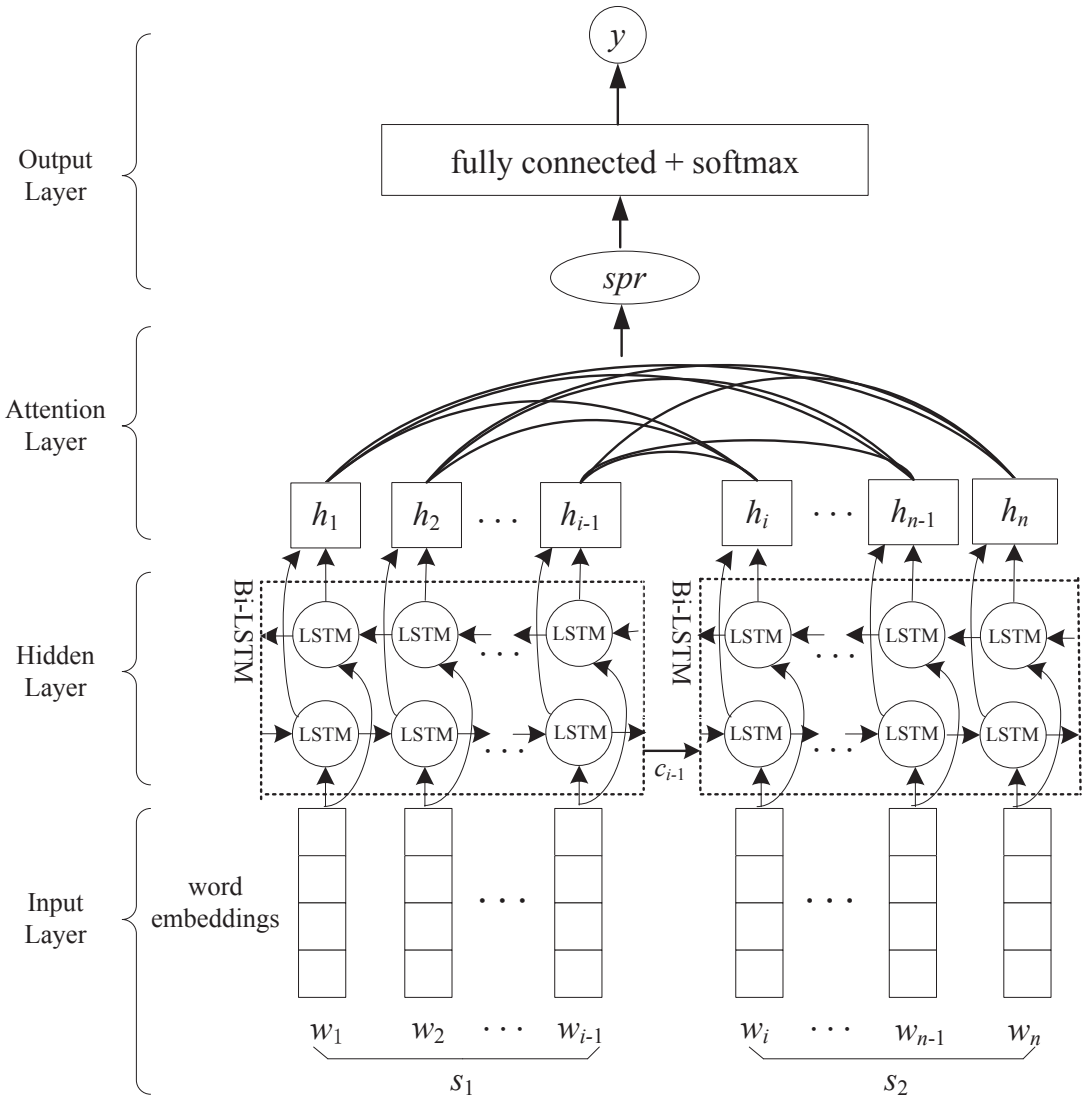

FIGURE 1. The neural network architecture

the resulting representations showcase interesting linear substructures of the word vector space. We use the 300-dimensional GloVe word embeddings trained on 840 billion tokens. We also use 25-dimensional PARAGRAM vectors since they were developed for paraphrase tasks, having been trained on word pairs from the Paraphrase Database. We concatenate on two word embeddings and obtain 325-dimensional vectors for each input word. Phrase vectors and sentence vectors are generated from word vectors. The pre-training word embeddings are considered to contain certain latent semantic information. Semantically related words have higher vector similarity. The addition and subtraction operations between the word vectors can be performed to obtain the semantic relationship between the words. Therefore, using pre-training word embeddings as input features can more naturally display the potential grammar and semantic information between words, which has positive effect on semantic similarity.

In this paper, we use regular expressions to tokenize input in the training set and return a list of all words and delimiter tokens. The vocabulary V is created according to the list. The pre-training word vector file is read and an embedding dictionary containing key-value pairs {word: word vector} is created. A look-up table T is generated by the get() method to return the corresponding value of the specified key in the embedding

dictionary V, where $|V|$ is the vocabulary size and d is the dimension of the word vector. For out-of-vocabulary words in the training set, we use a random function to initialize an array of dimension d and populate it with random samples from a uniform distribution over $(0, 1)$. Given the input sentence $s = \{w_1, w_2, \ldots, w_T\}$, the sentence is converted to feature sequences by mapping each word to an index in the look-up table.

3.2. **Conditional Bi-LSTM encoding.** Recurrent neural networks with long short-term memory units have been successfully applied to a wide range of NLP tasks, such as machine translation, speech recognition, and question answering [16,17]. LSTMs encompass memory cells that can store information for a long period of time, as well as three types of gates that control the flow of information into and out of these cells: input gates i, forget gates f and output gates o. LSTM memory cell with gating units is shown in Figure 2.

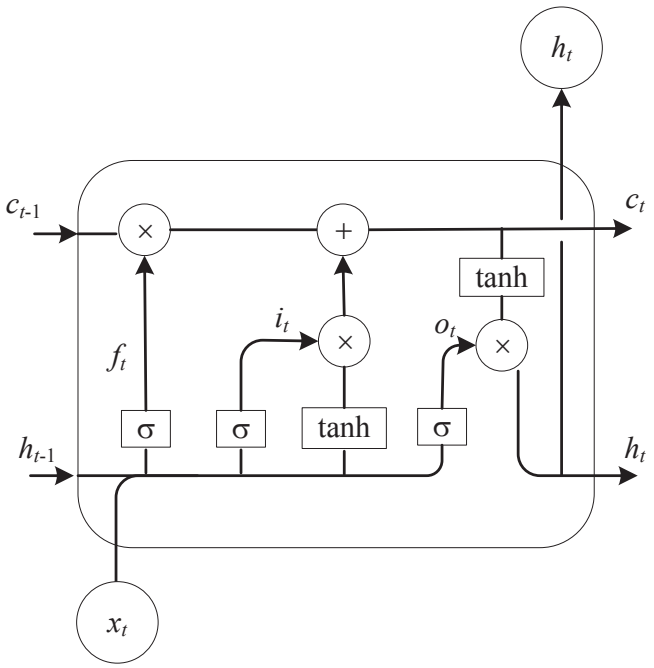

FIGURE 2. LSTM memory cell with gating units

Given an input vector x_t at time step t, the previous output h_{t-1} and cell state c_{t-1}, an LSTM with hidden size k computes the next output h_t and cell state c_t as

$$H = \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} \tag{1}$$

$$i_t = \sigma\left(W^i H + b^i\right) \tag{2}$$

$$f_t = \sigma\left(W^f H + b^f\right) \tag{3}$$

$$o_t = \sigma\left(W^o H + b^o\right) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W^c H + b^c\right) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where W^i, W^f, W^o, $W^c \in \mathbb{R}^{2k \times k}$ are trained matrices and b^i, b^f, b^o, $b^c \in \mathbb{R}^k$ are trained biases that parameterize the gates and transformations of the input, σ denotes the element-wise application of the sigmoid function and $\odot$ denotes the element-wise multiplication of two vectors. It can be found from the above formulas that i_t, f_t and o_t are all probability

values between 0 and 1, which are used to describe the degree the memory cell update, forgetting and filtering. The gate mechanism encourages information to be filtered, so that the information that should be saved is always passed and the information that should not be memorized is intercepted by the gates.

LSTM can only capture information from the past sequence as well as the current input. However, the predicted results of many tasks depend on the entire input sequence. Bi-LSTM connects two hidden layers of opposite directions to the same output. With this form of generative deep learning, the output layer can get information from past (backward) and future (forward) states simultaneously. Two directional neurons do not have any interactions.

We propose a conditional Bi-LSTMs encoder, which converts the input sentence-pair into a vector of fixed length. The corresponding outputs of the forward and backward in the Bi-LSTM are added as the final hidden state output at each time step. The proposed model can mine richer semantic features and have stronger representational ability because directional recurrent neural networks can capture the past and future information of the whole sequence at the same time. In contrast to learning sentence representations, we are interested in neural models that read both sentences to determine semantic relations, thereby reasoning over equivalences or contradictions of pairs of words, phrases and sentences. The first sentence is read by a Bi-LSTM. A second Bi-LSTM with different parameters is reading the second sentence, but its memory state is initialized with the last cell state (c_{i-1} in Figure 1) of the previous Bi-LSTM, i.e., it is conditioned on the representation that the first Bi-LSTM was built for the first sentence. Dropouts are applied outside the cells of LSTMs, which can better control the data flow of each gate of LSTM memory cells and reduce overfitting on semantic similarity task. We pad sequences to the same length since the lengths of the input sentences are not equal. Let L be the maximum length of the input sentence. Sequences that are shorter than L are padded with 0 at the end. Sequences longer than L are truncated so that they fit the desired length.

3.3. **Soft-alignment attention mechanism.** Recurrent neural network is considered to transmit the past state information to the current time step but in reality the ordinary RNN structure is difficult to pass information separated by a very long distance. For example, the hidden memory cells at the last time step can get information of the whole sentence in theory. However, the information carried in the previous input will be diluted by the subsequent sequence for a long input in practice. There is very little information about the first time step in the last hidden layer output. To solve the above problem, an attention mechanism is used to learn weights for the hidden layer output at each time step, which allows the model to attend over the past output vectors that can make decisions for classifier. The attention mechanism performs weighted data transformation on the source data sequence, which enables the task processing system to focus more on the significant information related to the current output in the input data and improves the quality of the output. The key aspect of attention mechanism is that the weighting and filtering of data are not defined by human which are automatically learned from a large number of temporal structural relationships.

For determining whether one sentence is semantically equivalent to another, it can be a good strategy to check for semantic equivalence or contradiction of individual word and phrase pairs. To encourage such behavior, we propose a soft-alignment attention mechanism which can obtain a sentence-pair encoding from fine-grained reasoning via soft-alignment of words and phrases in the sentence-pair. Let Y be a matrix consisting of output vectors $[\, h_1 \;\; h_2 \;\; \ldots \;\; h_i \;\; \ldots \;\; h_L \,]$ $(i \in [1, L])$ that the first Bi-LSTM produced when

reading the first sentence, where k is a hyperparameter denoting the size of embeddings and hidden layers, L is the maximum length of the input sentence and h_i is the output vector at the ith time step. Our model attends over the first Bi-LSTM's output vectors of the first sentence while the second Bi-LSTM processes the second sentence one word at a time, and learns semantic similarity scores on all output vectors of the first sentence for every word w_t with $t \in (L+1, N)$ in the second sentence. The semantic similarity scores for every word w_t are shown in Figure 3.

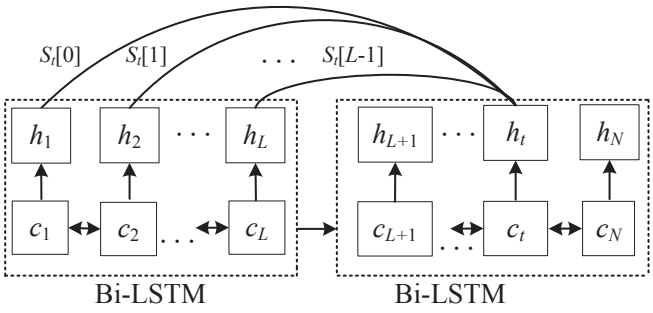

FIGURE 3. Semantic similarity scores for every word w_t

As shown in Figure 3, by learning the relationship between the contextual representation of the first sentence and the output vector at the ith time step, the network can inform the second Bi-LSTM which text of the first sentence is most relevant to the content being described. More precisely, a Bi-LSTM with attention for semantic similarity task does not need to capture the whole semantics of the first sentence in its cell state. Instead, it is sufficient to output vectors while reading the first sentence and accumulating a representation in the cell state that informs the second Bi-LSTM which of the output vectors of the first sentence it needs to attend over to determine the class (0 or 1).

A higher score means that the corresponding pair of text is given more attention. The following formula is used to learn the semantic similarity scores for word w_t.

$$S_t = \tanh\left(W^y Y + W^h h_t \otimes e_L\right) \tag{7}$$

where W^y, $W^h \in \mathbb{R}^{k \times k}$ are trained projection matrices, e_L is a vector of 1s and $h_t \otimes e_L$ is repeating the linearly transformed h_t as many times there are words in the first sentence.

The attention weights vector is calculated as follows.

$$\alpha_t = \text{softmax}\left(W^T S_t\right) \tag{8}$$

The weighted representation r_t of the first sentence for every word w_t can be modeled as follows.

$$r_t = Y\alpha_t^T + \tanh\left(W^r r_{t-1}\right) \tag{9}$$

Note that r_t is dependent on not only all the weighted output vectors of hidden layer at the current time step, but also the previous attention representation r_{t-1} to inform the model about what was attended over in the previous step. The semantic matching information at each time step in the time series is saved and passed to the next time step. The last weighted semantic vector contains the global matching information for the sentence-pair. The final sentence-pair representation *spr* is obtained from a non-linear combination of the last attention-weighted representation r_N and the last output vector h_N integrated global context information.

$$spr = \tanh\left(W^x h_N + W^r r_N\right) \tag{10}$$

where W^r, $W^x \in \mathbb{R}^{k \times k}$, $spr \in \mathbb{R}^k$.

4. Experiments and Results.

4.1. Datasets and evaluation metrics.

We evaluate our model on two tasks.

i) Paraphrase identification: given a pair of sentences, we predict a binary label indicating whether the two sentences are paraphrases. Microsoft Research Paraphrase Corpus is used for this task. It includes 5,801 pairs of sentences extracted from news source on the web, with 4,076 (2,753 true, 1,323 false) for training and the remaining 1,725 (1,147 true, 578 false) for testing. This task is evaluated by accuracy and $F1$. Accuracy is the proportion of true results among the total number of samples, which is a statistical measure of how well a binary classification test correctly identifies. $F1$ considers both the precision and the recall to compute the score, where precision is the number of correct positive results divided by the number of all positive results returned by the classifier, and recall is the number of correct positive results divided by the number of all relevant samples. The $F1$ score is the harmonic average of the precision and recall.

ii) Semantic relatedness measurement: given a pair of sentences, we measure a semantic similarity score of this pair. Sentences Involving Compositional Knowledge dataset provided in SemEval-2014 task 1 is used for this task. It consists of 9,927 sentence pairs, with 4,500 for training, 500 as a development set, and the remaining 4,927 in the test set. The sentences are drawn from image and video descriptions. Each sentence pair is annotated with a relatedness score $\in [1, 5]$, with higher scores indicating the two sentences are more closely-related. This task is evaluated by Pearson's r, Spearman's ρ and Mean Squared Error (MSE). The first two metrics measure the degree of correlation between the prediction and the annotation. Pearson's correlation assesses linear relationships and Spearman's correlation assesses monotonic relationships. The MSE is a measure of the quality of an estimator. It is always non-negative, and values closer to zero are better.

4.2. Experiment settings.

For classification, a softmax layer is applied over the output of a non-linear projection of the sentence-pair representation into the target space of the two classes (0 or 1) using

$$y = \operatorname{argmax}\left(\operatorname{softmax}\left(W^s spr\right)\right) \tag{11}$$

where spr denotes the sentence-pair representation, $W^s \in \mathbb{R}^{2k\times 2}$.

We use the cross entropy loss with L2 regularization for the paraphrase identification task. The training objective is to minimize the following cost function:

$$J = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log\hat{y}^{(i)} + \left(1-y^{(i)}\right)\log\left(1-\hat{y}^{(i)}\right)\right] + \frac{\lambda}{2n}\|\theta\|_2^2 \tag{12}$$

where $y^{(i)}$ and $\hat{y}^{(i)}$ denote the ground truth label and the predicted value of the ith sample, n is the number of training examples, θ is the model weight vector to be trained, and λ is the regularization parameter.

To compute a similarity score of a pair of sentences for SICK dataset, we replace *classes* $= 5$ and use the following equation to calculate a predicted similarity score $\hat{y}$.

$$\hat{p}_\theta = \operatorname{softmax}\left(W^s spr\right) \tag{13}$$

$$\hat{y} = r^T \hat{p}_\theta \tag{14}$$

where spr denotes the sentence-pair representation, $W^s \in \mathbb{R}^{2k\times 5}$, $\hat{p}_\theta$ is the predicted distribution with model weight vector θ, $r^T = [1\ \ 2\ \ 3\ \ 4\ \ 5]$.

A sparse target distribution p which satisfies $y = r^T p$ is computed as:

$$p_j = \begin{cases} y - \lfloor y \rfloor, & i = \lfloor y \rfloor + 1 \\ \lfloor y \rfloor - y + 1, & i = \lfloor y \rfloor \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $j \in [1, 5]$, and y is the similarity score.

We use regularized KL-divergence loss for the semantic relatedness task. The training objective is to minimize the KL-divergence loss plus an L2 regularizer:

$$J = -\frac{1}{n} \sum_{i=1}^{n} KL\left(p^{(i)} \| \hat{p}_\theta^{(i)}\right) + \frac{\lambda}{2n} \|\theta\|_2^2 \tag{16}$$

where n is the number of training pairs, θ denotes the model parameters, $p^{(i)}$ and $\hat{p}_\theta^{(i)}$ denote the target distribution and the predicted distribution of the ith sentence-pair.

The Adam optimization algorithm is used to dynamically calculate adaptive learning rates for different parameters. For every task, we perform a small grid search over the following settings and find the optimal combination of parameters: LSTM hidden units $[50, 80, 100, 150, 200]$, dropout $[0.1, 0.2, 0.3, 0.4]$, L2-regularization strength $\lambda = 0.03$, and mini-batch size = 25.

4.3. **Attention visualization.** It is instructive to analyze which output representations the model is attending over when deciding the class of a paraphrase identification example. In the following we visualize and discuss the attention pattern of the presented attentive model. Two pairs of sentences are hand-picked from the positive and negative samples of the training set. Attention visualizations for the positive and negative examples are depicted in Figure 4 and Figure 5. The visualization results are graphical representations of data where the individual values contained in a matrix are represented as colors. Larger values are represented by darker squares and smaller values by lighter pixels.

In the first positive example, it can be seen that attentions are mainly focused on the same pairs of words, such as "Paracha" and "Paracha", "faces" and "face", "person" and "person", and "convicted" and "convicted". We find that the proposed attention

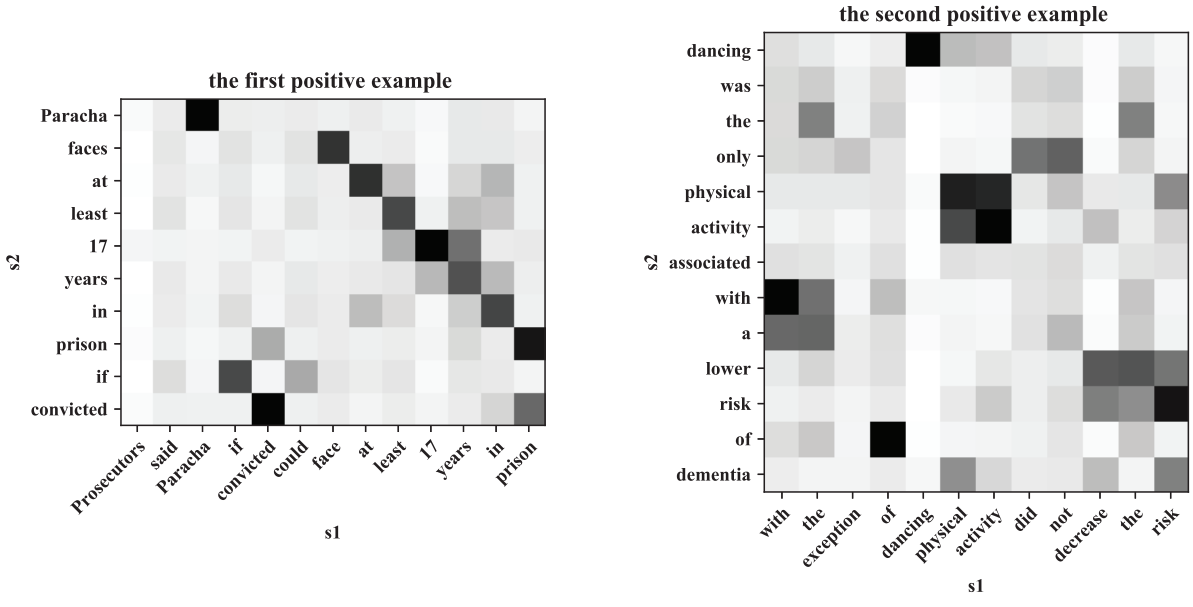

FIGURE 4. Attention visualization for the positive examples

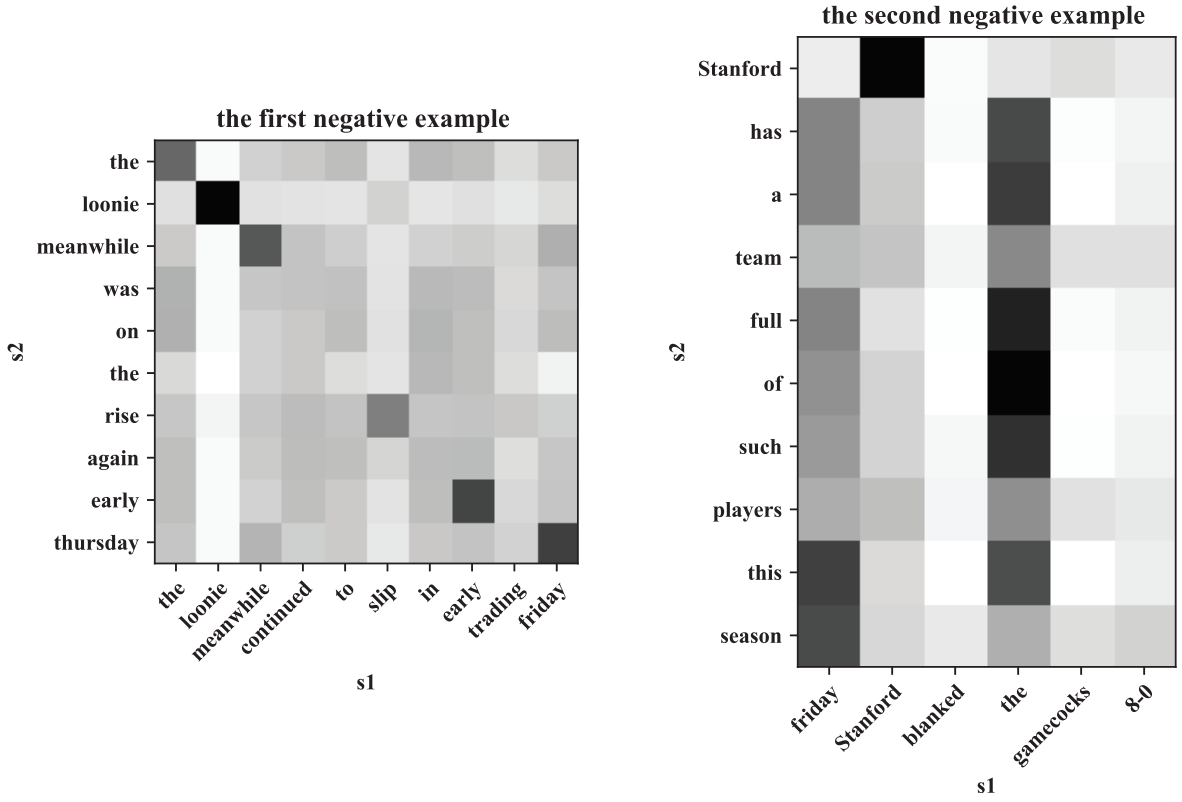

FIGURE 5. Attention visualization for the negative examples

can easily detect if the sentence is simply a reordering of words in another sentence. In the second positive example, it can be found that the words or phrases "dancing" and "physical activity" appearing in both sentences are attended. In addition, the phrase-pairs "with a" and "with the", "lower risk" and "decrease the risk" have received more attention, which shows that the model is able to resolve synonyms in the text and capable of matching the same meaning expressions. The attentions gained by "dementia" and "physical", "only" and "did not" indicate that the proposed attention mechanism seems to also work well when words in the sentence pair are connected via deeper semantics or common-sense knowledge.

It is worth noting that "thursday" and "friday", "rise" and "slip" get relatively more attention in the first negative example, which shows that the model pays attention to the pairs of the words that are semantically contradictory and makes decision for the classifier. In the second negative example, the attentions are focused on the function word "the" except for the common word "Stanford". The reason is that the two sentences are entirely unrelated. In such cases, the sentence-pair representation is likely dominated by the last output vector instead of the attention weighted representation.

4.4. **Results on two datasets.** We report accuracies and $F1$ scores from prior works in Table 1. Lexicalized classifier [18] constructed features from the BLEU score between the sentence-pair, length difference, word overlap, uni- and bigrams, part-or-speech tags, as well as cross uni- and bigrams. Finch et al. [5] and Madnani et al. [3] used the machine translation evaluation metrics as the classification features. When comparing to the conventional models required hand-crafted features, we outperform them by 1.4 percentage points in accuracy and 0.7 percentage points in $F1$.

TABLE 1. Test set results on MRPC for paraphrase identification

Category	Model	Acc.	$F1$
Hand engineered features	Lexicalized classifier	69.8	80.4
	Finch et al. (2005)	74.9	82.6
	Madnani et al. (2012)	77.4	84.1
Neural network models	ARC-I (2014)	69.6	80.3
	ARC-II (2014)	69.9	80.9
	Bi-CNN-MI- (2015)	72.5	81.4
	Bi-CNN-MI (2015)	78.1	84.4
	He et al. (2015)	78.6	84.7
	MV-LSTM (2017)	75.4	82.8
	Match-SRNN (2017)	74.5	81.7
	Open AI GPT (2018)	–	82.3
	GLUE (2018)	77.3	83.5
	M-MaxLSTM-CNN (2018)	78.1	84.5
This work	random initialization embeddings	70.3	81.3
	pre-training embeddings	75.1	82.7
	pre-training embeddings, conditional encoding	76.3	83.5
	pre-training embeddings, conditional encoding, attention	**78.8**	**84.8**

Approaches shown in the third rows of Table 1 are based on neural networks [1,2,4,12,19-21]. The Bi-CNN-MI model presented by Yin and Schutze [4] learns multi-granular sentence representations using CNN and compares interaction features at each level. The model includes a pre-training technique which significantly improves results, as shown in the table. We do not use any pre-training but still outperform their best results. The accuracy of MV-LSTM model based on multiple positional sentence representations and the recursive matching structure Match-SRNN on MRPC was 75.4% and 74.5% respectively [19]. Open AI explored a GPT (Generative Pre-training) approach [20] for language understanding tasks using a combination of unsupervised pre-training and supervised fine-tuning. Their training procedure consisted of two stages. The first stage was learning a high-capacity language model on a large corpus of text. This is followed by a fine-tuning stage, where they adapted the model to a discriminative task with labeled data. The $F1$ score of GPT on MRPC is 82.3%, which is 2.5 percentage points lower than our model. We also achieve improvement of 1.3% in $F1$ on the recently introduced GLUE (General Language Understanding Evaluation) multi-task benchmark [21]. The best previous neural network model on MRPC was from He et al. [2], which modeled each sentence using a CNN that extracted features at multiple levels of granularity and used multiple types of pooling. He et al. used multiple kinds of embeddings including 300-dimensional GloVe embeddings, 25-dimensional PARAGRAM embeddings and 200-dimensional part-of-speech embeddings to represent each sentence. We do not use part-of-speech embeddings. Our model still outperforms He et al's performance in accuracy and $F1$, which received the best result in the all neural network models.

From the comparison in Table 1, we can also see the improvement of our work over the best method in the literature is not obvious. We analyze the reasons as follows. First, the model does not use any other language features expect for the pre-training word embeddings. The performance of the model will be further improved if the other language features such as part-of-speech and named entity annotation are added to the input layer. Second, there is a loss of information when the sentence is compressed by

the encoder to a vector of fixed length. How to effectively reduce the information loss in the process of sentence compression and bring more detailed semantic information to the classifier is an important content of future research.

We identify three major components of our model: pre-training word embeddings, conditional Bi-LSTM encoding, attention mechanism, and evaluate their performances for paraphrase identification task. The experimental results are also reported in Table 1. It can be seen that using the embeddings, which are trained in advance on unlabeled large-scale corpora, instead of using random initialization embeddings gives an improvement of 4.8 percentage points in accuracy. It demonstrates that using pre-training word embeddings can provide a significant performance boost. We observe that processing the second sentence conditioned on the first sentence instead of encoding each sentence independently gives an improvement of 1.2 percentage points in accuracy. We argue this is due to information being able to flow from the part of the model that processes the first sentence to the part that processes the second sentence. Specifically, the model does not waste capacity on encoding the second sentence, but can read the second sentence in a more focused way by checking words and phrases for equivalences or contradictions based on the semantic representation of the first sentence. Enabling the model to attend over output vectors of the first sentence for every word in the second sentence yields another 2.5 percentage point improvement in accuracy compared to attending based only on the last output vector of the first sentence. We argue that this is due to the model being able to check for equivalence or contradiction of individual words and phrases in the second sentence. Reasoning over equivalence or contradiction of pairs of words and phrases provides an effective decision for the classifier.

Our results on the SICK are summarized in Table 2, showing Pearson's r, Spearman's ρ, and MSE. The first three models [22-24] were submitted from SemEval-2014 competition. The models relied heavily on feature engineering and measured the semantic relatedness between sentences by combining features such as word overlap, part-or-speech tags, syntax and word distance. Other results on the SICK dataset were from the representative models [2,12,25] since 2015. Dependency Tree-LSTM presented by Tai et al. [25] requires a dependency parser. On the contrary, our approach does not rely on parse trees. When measured by Pearson's r, the previous state-of-the-art approach was M-MaxLSTM-CNN presented by Tien et al. [12], which used a CNN to learn a multi-aspect word embedding from various pre-training word embeddings and applied the max-pooling scheme and LSTM on the embedding to forming a sentence representation. Spearman's ρ and MSE results of the M-MaxLSTM-CNN model on the SICK dataset were not reported. Our model does not use hand-crafted features and does not require pre-training technique, which achieves state-of-the-art on SICK dataset.

Table 2. Test set results on SICK for semantic relatedness

Model	r	ρ	MSE
UNAL-NLP (2014)	0.8043	0.7458	0.3593
Meaning Factory (2014)	0.8268	0.7722	0.3224
ECNU (2014)	0.8295	0.7689	0.3250
Constituency Tree-LSTM (2015)	0.8582	0.7966	0.2734
Dependency Tree-LSTM (2015)	0.8676	0.8083	0.2532
He et al. (2015)	0.8686	0.8047	0.2606
M-MaxLSTM-CNN (2018)	0.8876	–	–
This work	**0.8993**	**0.8151**	**0.2529**

5. **Conclusions.** We presented the recurrent neural network architecture for semantic similarity. Based on the insight that semantic similarity requires paying attention to text interaction matching patterns and comparing two sentences on multiple levels of granularity, we learn sentence representations and reason semantic equivalence or contradiction of pairs of words, phrases and sentences using conditional Bi-LSTMs encoding and soft-alignment attention mechanism. We have demonstrated that the benefit of using combined pre-training word embeddings and the effectiveness of our model on paraphrase identification and semantic relatedness tasks. Results on MRPC and SICK are state of the art in all the neural network models. In the future, we plan to extend this model to other natural language processing tasks including question answering and copy detection.

Acknowledgment. This work is supported by National Key Research and Development Project of China (No. 2017YFC0821003-3), Scientific Research Fund of the Higher Education Institutions of Liaoning Province, China (No. 2017J049) and Natural Science Foundation of Liaoning Province of China (No. 20180550395). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] B. Hu, Z. Lu, H. Li and Q. Chen, Convolutional netural network architectures for matching natural languages sentences, *Proc. of Advances in Neural Information Processing Systems*, Montreal, Canada, pp.2042-2050, 2014.

[2] H. He, K. Gimpel and J. Lin, Multi-perspective sentences similarity modeling with convolutional neural networks, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp.1576-1586, 2015.

[3] N. Madnani, J. Tetreault and M. Chodorow, Re-examining machine translation metrics for paraphrase identification, *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, pp.182-190, 2012.

[4] W. Yin and H. Schutze, Convolutional neural network for paraphrase identification, *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, CO, pp.901-911, 2015.

[5] A. Finch, Y. Hwang and E. Sumita, Using machine translation evaluation techniques to determine sentence-level semantic equivalence, *Proc. of the 3rd International Workshop on Paraphrasing*, Jeju Island, Korea, pp.17-24, 2005.

[6] L. Qiu, M. Kan and T. Chua, Paraphrase recognition via dissimilarity significance classification, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp.18-26, 2006.

[7] S. Hassan, *Measuring Semantic Relatedness Using Salient Encyclopedic Concepts*, Ph.D. Thesis, University of North Texas, Denton, TX, USA, 2012.

[8] A. Eyecioglu and B. Keller, Twitter paraphrase identification with simple overlap features and SVMs, *Proc. of the 9th International Workshop on Semantic Evaluation*, Denver, USA, pp.64-69, 2015.

[9] W. Xu, A. Ritter and C. Chris, Extracting lexically divergent paraphrases from Twitter, *Transactions of the Association for Computational Linguistics*, vol.2, pp.435-448, 2014.

[10] Z. Lu and H. Li, A deep architecture for matching short texts, *Proc. of the 27th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, USA, pp.1367-1375, 2013.

[11] J. Gao, P. Pantel and M. Gamon, Modeling interestingness with deep neural networks, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp.2-14, 2014.

[12] H. N. Tien, M. N. Le and Y. Tomohiro, Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity, *The Computing Research Repository*, 2018.

[13] W. Lan and W. Xu, Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering, *Proc. of the 27th International Conference on Computational Linguistics*, NM, USA, pp.3890-3902, 2018.

[14] J. Pennington, R. Socher and C. Manning, GloVe: Global vectors for word representation, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp.1532-1543, 2014.

[15] J. Wieting, M. Bansal and K. Gimpel, From paraphrase database to compositional paraphrase model and back, *Transaction of the Association for Computational Linguistics*, vol.3, pp.345-358, 2015.
[16] K. Shimoyama, A. Ueno and T. Takubo, An evaluation of web article similarity based on user comments, *ICIC Express Letters*, vol.13, no.5, pp.409-413, 2019.
[17] P. M. Tasinaffo and A. R. Neto, Adams-Bashforth neural networks applied in a predictive control structure with only one horizon, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.445-464, 2019.
[18] Z. Kozareva and A. Montoyo, Paraphrase identification on the basis of supervised machine learning techniques, *Proc. of the International Conference on Natural Language Processing*, Turku, Finland, pp.524-533, 2006.
[19] L. Pang, Y. Lan and J. Xu, A survey on deep text matching, *Chinese Journal of Computers*, vol.40, no.4, pp.985-1003, 2017.
[20] A. Radford, K. Narasimhan and T. Salimans, *Improving Language Understanding by Generative Pre-training*, https://blog.openai.com/language-unsupervised/, 2018.
[21] A. Wand, A. Singh and J. Michael, GLUE: A multi-task benchmark and analysis platform for natural language understanding, *arXiv preprint arXiv:1804.07461*, 2018.
[22] S. Jimenez, D. George and J. Baquero, UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment, *Proc. of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.732-742, 2014.
[23] J. Bjeva and B. Johan, The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity, *Proc. of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.642-646, 2014.
[24] J. Zhao, T. Zhu and M. Lan, ECNU: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textural entailment, *Proc.of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.271-272, 2014.
[25] K. S. Tai, R. Socher and C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks, *Proc. of the Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Beijing, China, pp.1556-1566, 2015.