Sentence-Embedding and Similarity via Hybrid Bidirectional-LSTM and CNN Utilizing Weighted-Pooling Attention

Degen HUANG†‡, Anil AHMED†§, Syed Yasser ARAFAT††, Khawaja Iftekhar RASHID††, Qasim ABBAS†††, Nonmembers, and Fuji REN††††, Fellow

SUMMARY Neural networks have received considerable attention in sentence similarity measuring systems due to their efficiency in dealing with semantic composition. However, existing neural network methods are not sufficiently effective in capturing the most significant semantic information buried in an input. To address this problem, a novel weighted-pooling attention layer is proposed to retain the most remarkable attention vector. It has already been established that long short-term memory and a convolutional neural network have a strong ability to accumulate enriched patterns of whole sentence semantic representation. First, a sentence representation is generated by employing a siamese structure based on bidirectional long short-term memory and a convolutional neural network. Subsequently, a weighted-pooling attention layer is applied to obtain an attention vector. Finally, the attention vector pair information is leveraged to calculate the score of sentence similarity. An amalgamation of both, bidirectional long short-term memory and a convolutional neural network has resulted in a model that enhances information extracting and learning capacity. Investigations show that the proposed method outperforms the state-of-the-art approaches to datasets for two tasks, namely semantic relatedness and sentence similarity, sentence embedding, deep learning, long short-term memory, convolutional neural network

1. Introduction

In the present era, text understanding and information retrieval research have generated great interest among researchers for analyzing the sentence similarity in natural language processing (NLP). Although computing similarity facilitates many tasks such as plagiarism detection [1], query rank [2], and question answering [3], yet it is still a challenging issue, largely due to the intrinsic ambiguity of various linguistic sequences and the inadequate number of labeled training corpora available. On the other hand, conventional NLP methodologies utilized to operate similarity systems strongly depend on hand-crafted features [4]. Because a word is embedded in a multi-dimensional space with high-dimensional vectors, these features suffer from sparsity, unavoidably resulting in unacceptable complexity and inefficiency.

Recent advancements in deep neural network (DNN) models have already brought outstanding performance improvements in various artificial intelligence operations related to computer vision, speech recognition, robotics, and NLP. In the last decade, many researchers have presented the effective modeling of words referred to as ‘distributed word embedding’, which are competent to explicitly encode both the semantic and syntactic information [5]. In this method, a word is embedded in the multi-dimensional vector space, where similar semantics contexts lie close to each other, e.g., the vector (“Madrid” – “Spain”) + (“France”) is closer to vector (“Paris”) than any other word vector.

Today, word embedding has become extremely effective and is applied to various applications such as part-of-speech (POS) tagging [6] and name entity recognition [7]. It has also demonstrated the potential for neural word representations for analogies tasks.

Moreover, the increasing number of applications of DNNs has diverted the attention from word-level to larger texts, such as sentences embedding. Earlier approaches applied to look for sentence similarity are only able to capture the brief dependencies and similarities among characters, which is indeed not appropriate for finding similarity accurately. Therefore, these traditional approaches have limitations in capturing rich input sequence patterns between their correlations. In short, sentence embedding plays a vital role and is an emerging fundamental parameter in computing similarity.

Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) are the two main types of neural network models, which are widely used in NLP because of their ability to capture the long-term and local dependencies, respectively. In machine translation, the RNN with LSTM block along with another Gated Recurrent Unit (GRU) have been proposed recently, each having the capability to encode a given source linguistic sequence. Due to their design, these models are able to capture richer simple and comprehensive semantic representation information hidden in long text sentences. A trained RNN-LSTM model, weakly supervised sequentially, takes each word from a given sentence sequence, extracts its richer information and embeds it into a
vector consisting of semantic meaning, whereas the similarity can be computed using the cosine distance between two vectors [8]. Originally, the Convolutional Neural Network (CNN) was at first conceived for computer vision tasks [9], however, it also exhibited promising outcomes [10] in extracting the semantic features of short sentences and the sentence modeling. A standard CNN is usually consists of several convolutional layers, dynamic k-max pooling and fully connected layers over linear sequences. In this study, the convolution filters are employed first to extract maps of basic features. Later, a pooling scheme is applied to feature maps to extract the most relevant features. Next, an output of all feature vectors is concatenated in the form of a fixed length single feature vector, also called a temporal vector. Temporal vectors are passed through LSTMs and finally used as an input to feed the attention mechanism. The attention mechanism has been drawn considerable interest in recent years due to its better internal representations with highly parallelizable computation, significantly shorter training time and its flexibility in modeling dependencies [11].

All of the existing approaches for capturing a text’s internal representations, including both semantic and syntactic, discards important information to some extent. This is also the case with the Principal Component Analysis (PCA)[12], where the lost information could have been beneficial for the extraction of some significant semantic information. To surmount the weakness of previous approaches, the new weighted-pooling attention (WPA) mechanism has been developed. We have also employed a bidirectional LSTM model to boost the information extraction competency. The LSTM model has also been combined with the one-dimensional CNN structure to capture the entire context information.

The prominent contributions of our research are as follows:

- A novel siamese structure [13] is devised consisting of two sub-networks bidirectional-LSTM (BLSTM), and a hybrid of LSTM-CNN. The combination of both models upsurges the ability of a model to extract comprehensive contextual information.
- A WPA mechanism is proposed for sentence level embedding, enhancing the scope from word-level features to a sentence-level, retaining the most significant information, order pattern and ignoring irrelevant words.
- The proposed model was evaluated by using different evaluation metrics, consisting of two distinct type tasks: the first is the SemEval 2014 semantic relatedness task-1 [14] and the second is the Microsoft research paraphrase identification task-2 [15]. The obtained empirical results obtained have achieved high efficiency and accuracy as compared with existing state-of-the-art approaches.

The remainder of the article is organized as follows: First, a brief overview of related work is given in Sect. 2. Next, the proposed model is described with detail in Sect. 3. Later, the experimental results with performance evaluation of the proposed method are compared with state-of-the-art approaches in Sect. 4. Finally, conclusions are drawn and future directions indicated in Sect. 5.

2. Related Work

Numerous efforts have been made with respect to sentence embedding and similarity computation. Earlier research is primarily interested in featured engineering. Mihalcea et al. proposed a corpus-based and knowledge-based method for measuring the semantic similarity of short texts [16]. The recent rise of deep learning models is interested in exploring new and efficient ways to learn continuous vectors representing words/sentences; later researchers called it “word embedding”.

In the word embedding methods, Continuous Bag-of-Words (CBOW) and the skip-gram model have favorably enhanced sentence embedding by utilizing a local attention mechanism [17]. The CBOW performs extremely well on a variety of NLP tasks, but it has the drawback of missing word order patterns, which is the most key factor in the semantic sentence evaluation.

Inspired by the word embedding method, the deep-structured semantic model [18] and convolutional latent semantic model [19] have been developed for information retrieval, wherein a word sequence is discovered by the model; thus it can also be interpreted as a sentence embedding method. Gan et al. [20] propose a new type of hierarchical CNN-LSTM architecture for modeling sentences based on encoder and decoder modules. The CNN is used as an encoder to map an input sentence into a continuous representation, and decoding is performed through LSTM. A proposed model called Manhattan (Ma) LSTM, based on two networks LSTMa and LSTMb, describes that, given a large corpus, the basic LSTMs have the capability to model their complex semantics [21]. The recursive neural network (ReNN) utilizes only the structure of parse trees, which are used to transform word vectors into sentence vectors [22]. The learning ability of ReNN relies deeply on the construction of the tree vector, where the tree construction is complex and time-consuming. RNN is a special case of the ReNN, and both models have been successfully combined with word vector representations to predict the semantic similarity and phrase pairs [23]. Another proposed model is a hybrid model of CNN and the LSTM network, consisting of a pairwise word interaction model and similarity focused layer to capture semantic information [24]. It has achieved satisfying results on three SemEval tasks. A CNN based on two architectures (ARC–I and ARC–II) reveals the superior power to perform sentence modeling with their rich matching patterns through layer-by-layer composition and pooling operation [25].

The order-insensitive model has achieved very good performance on a variety of tasks, but it suffers the weakness of insufficiency to fully capture the semantics of natural language. To overcome this problem, Tai et al. [26]
introduced a tree-structured LSTM (TS-LSTM) model implemented on two tasks: semantic relatedness and sentiment classification. It consists of three main units: input vector, hidden states and the multiple arbitrarily child units. Mostly DNN-based methods have the complex hierarchical structure which causes an increase in training time. A recently reported proposal to reduce the training time utilizes shallow CNN model skipping fully connected layers [27]. The proposed CNN [28] is based on two-component frameworks: first, sentence modeling is performed through a convolution feature detector having multiple window sizes and pooling, and later, a similarity measurement is compared with the local region of vector representations using various measurement functions like cosine distance, Euclidean distance, and element-wise difference.

3. The Proposed Model

In this section, the proposed model framework is discussed in detail along with its main modules. The main modules are illustrated graphically in Fig. 1, which depicts two core module-based architectures to measure the degree of semantic similarity between two sentences. These modules are named as ARC-1 (based on BLSTM layer) and ARC-2 (based on hybrid layer consisting of both LSTM-CNN). In ARC-2, the model is applied to combine the outputs of the LSTM and CNN. The BLSTM and Hybrid LSTM-CNN framework is used to capture information independently contained in any position of a sentence. In CNN, convolutional filters perform convolutions operations to capture the local context information contained in an input sequence. Moreover, fully connected neural networks are utilized to separately integrate the outputs of the previous layer and forward into the WPA layer. A WPA layer is deployed to retain the most significant semantic information in the form of a final sentence representation inclusive of attention weights. Another main objective of our model is to reduce the distance among the embedding vectors of sentences for increasing similarity, where similar semantic meaning lies as close to each other as possible. Since LSTM-CNN have different features, the most well-known one is its ability to learn comprehensive contextual information.

The former method excels at representing sequential elements in a sentence, while the latter is good at modeling the entire sentence. Thus, it is an outstanding method to employ them in combination to extract word-level features to sentence-level. In the implementation here, the softmax function is applied as an activation function and the Adam [29] optimizer is applied to train this hybrid model.

3.1 Input Layer

It consists of $W^1, W^2, \ldots, W^T$ words input sequences, where $T$ is a total length of the given sentence.

$$S = \{ W^1, W^2, \ldots, W^T \}$$ (1)

3.2 Embedding Layer

Many approaches are considered for representing a word as a vector, such as the one-hot vector method, which achieves remarkable performance related to document classification tasks [30]. However, to classifying short sentences with this method leads to data sparsity or the curse of dimensionality. To overcome this, a distributional word embedding method has been proposed recently [31], which urges that continuous word representations are more suitable, wherein a word is represented by a dense vector. Numerous efforts have been made on pre-trained word embedding methods, trained entirely is an unsupervised manner on corpora under different models. The most well-known algorithms, such as Word2vec [31], GloVe [32], and fasttext [33], are capable of learning the semantic meaning of a given contextual window. Herein, GloVe distributional word embedding models are adapted to convert words into the real-valued vector. The main objective of the word vector model is to maximize the average log probability which is demonstrated in Eq. (2):

$$X_i = \frac{1}{T} \sum_{t=1}^{T-K} \log p(w_{i+k} | w_i)$$ (2)

$$p(y | w) = \frac{\exp(yw_T)}{\sum_i \exp(yw_i)}$$ (3)

Every word in Eq. (1) has been converted into a fixed size vector $em_i$. In the word embedding matrix following Eq. (4), where $W$ is a parameter that must be learned, where, $|V|$ is fixed-length vocabulary size and ‘d’ denotes the dimensionality of this embedding chosen by the user.

$$W \in \mathbb{R}^{d \times |V|}$$ (4)

Hence, we obtain a word embedding $em_i$ by using a matrix cross vector product, where $V^T$ is an integer value at index $em_i$ and 0 in all other positions and established for each input sentence.

$$em_i = W \times V^T$$ (5)

3.3 Bidirectional LSTM

Hochreiter et al. [34] present LSTM to overcome the problem of learning long-term dependencies called vanishing
At the current time step, and a former state \( h_t \) has been proposed for LSTM. Here, the Graves et al. model [35] consisted of input and output gates. It is a powerful standard model for sequential tasks because it captures a richer representation of input sentences. The main idea behind introducing an adaptive gating mechanism for LSTM is to keep the previous state memorizing and also to find a dense and low-dimensional representation. Many alternatives have been proposed for LSTM. Here, the Graves et al. model [35] is used, as depicted in Fig. 2. It consists of peephole connections from the constant error carousel (CEC) to the gates of the same memory block.

The key interest in using BLSTM is to take advantage of additional backward information exchange between the two LSTMs and thus, it enhances the memory capability. This helps the model to learn a better representation. The LSTM-based RNNs have four main compound components: one input gate \( I_t \) (Eq. (6)) also have corresponding weight matrix \( W_{xi}, W_{hi}, W_{ci}, b_i \); one forget gate \( F_t \) (Eq. (7)) with the corresponding weight matrix \( W_{xf}, W_{hf}, W_{cf}, b_f \); one output gate \( O_t \) (Eq. (8)) and corresponding weight matrix \( W_{xo}, W_{ho}, W_{co}, b_o \); all of those gates receive input from current input \( x_t \) at the current time step, and a former state \( h_{t-1} \) is generated. The current state of this cell \( c_{t-1} \) is called the peephole for decision making, and \( \sigma \) denotes the logistic sigmoid function and \( \odot \) denotes elementwise multiplication. The transition equations of LSTM are demonstrated as:

\[
I_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (6)
\]
\[
F_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (7)
\]
\[
O_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (8)
\]
\[
M_t = \tanh(W_{xm}x_t + W_{hm}h_{t-1} + W_{cm}c_{t-1} + b_m) \quad (9)
\]
\[
c_t = F_t \odot c_{t-1} + I_t \odot M_t \quad (10)
\]
\[
h_t = O_t \odot \tanh(c_t) \quad (11)
\]

Therefore, Eq. (10) \( c_t \) will be generated by pointwise multiplication of both the previous cell state and the current information. This model is mutually trained with all the other modules. During the training process, gradients of the cost function is computed by using the Adam optimization algorithm [29]. Fortunately, there are no external modules needed. Further, the model can be trained end-to-end.

### 3.4 Convolutional Neural Network

A powerful hierarchy of CNN has the ability to perform complex tasks. After empirical study [36] on deep neural architectures, it is suggested that multiple convolutional layers play a critical role in encoding significant semantic information. However, a simple CNN with little hyper-parameters also proves a state-of-the-art performance on several datasets in sentence classification. It consists of several convolutional, pooling and fully connected layers with a rectified linear activation function (ReLU). Both of the CNN-LSTM models are extensively effective for sentence modeling, and are especially suitable for sentence similarity. The study [37] establishes that the max-pooling outperforms the other pooling strategies in most domains. However, it has a drawback of missing position pattern, and it discards seemingly tenuous information, while that may be the most significant information. To tackle this problem, a novel WPA mechanism is presented. The proposed CNN applied in ARC-2 with a distinct sized window model framework is illustrated in Fig. 3. Sentence representations are formed by the CNN-LSTM and both are concatenated in the form of the feature vector as an input to the top WPA layer. In this study, the ability of the attention-based weighted-pooling mechanism to extract the important features is investigated using a hybrid of LSTM and one-dimensional CNN. The main reason for using hybrid LSTM-CNN is to increase their extraction capability for input data. As shown in Eq. (15) a standard CNN structure convolution is an operation between a vector of weights matrix \( M \in R^T \), where \( M \) is filter and input word sequence \( S \in R^T \) associated with the
single feature vector.

\[ C^T = f\left( \sum_{i \in M} S_i^{T-1} M^i + b^T \right) \]  

(15)

Today, ReLU (as defined in Eq. (16)) is the most commonly employed nonlinear activation function of CNN because it can reduce the number of iterations mandatory for convergence in deep learning networks. Here, a special variant of the ReLU called Leaky ReLU (LReLU) (as defined in Eq. (17)) is employed, which allows a small gradient when the unit is not active. It helps further to improve the learning efficiency as compared with ReLU.

The activation function \( f(x) \) of convolutional sub-layers is LReLU:

\[ f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0.01 & \text{if } x \leq 0 \end{cases} \]  

(16)

\[ f'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x \leq 0 \end{cases} \]  

(17)

3.5 Weighted-Pooling Attention

In machine translation, the attention mechanism was initially used in an encoder-decoder framework. The main incentive to introduce the attention is to allow the network to revisit all parts of a source sentence for an output decision, instead of trying to encode all information of a source sentence into a fixed-length vector. However, the attention mechanism has proved to be effective in finding the most discriminative features. A WPA mechanism based on BLSTM and hybrid LSTM-CNN is utilized in mining significant word features.

An attentive neural network with weighted-pooling is a data-driven framework to learn well-organized sentence representation from a given corpus[38]. Recently, it is of interest due to its highly parallelizable computation and the significantly lower training time it requires. The WPA is computed by Eq. (18), where \( y^T \) consists of the output vector \([y_1, y_2, y_3, \ldots, y_T]\) which is produced by a fully connected layer. The \( WP^T \) is a scalar value representing the corresponding weight at time step \( T \). It is computed similarly to the softmax function manner demonstrated in the following Eq. (19), where \( w^T \) is the vector parameters needed to be trained together with RNN using backpropagation. Whereas \( \exp(w^T y^T) \) is similar to the attention mechanism representing the potential focus at time step \( T \). If the occurrence of a word at time \( T \) has high significance, its weight will be high and therefore it gains high “attention”; if the value is low, the weight and “attention” will also be low. In this way, the model can learn to assign weights to different time steps from a given corpus. The suggested approach has the ability to learn from data and assign their weights in different time steps. If the values of all weights are same, weighted-pooling can be calculated by using the arithmetic mean.

\[ \text{AWP}^T = \sum_{T=1}^{n} WP_T y^T \]  

(18)

\[ WP_T = \frac{\exp(W^T y_T)}{\sum_{T=1}^{n} \exp(W^T y_T)} \]  

(19)

Numerous methods are utilized to measure the similarity between corresponding vector pairs, such as cosine distance, Euclidean distance, and Manhattan distance. Cosine similarity is opted for here in the designated model. It is a standard method for computing the cosine of the angle between two Euclidean vectors in the multi-dimensional space as defined in Eq. (21). The WPA weights will be learned during the training process and word embedding parameters are also adjusted.

\[ \cos \theta = \frac{x^T \cdot y^T}{\|x\| \|y\|} \]  

(21)

The entire computation process of this model is summarized as Algorithm 1 and 2:

Algorithm: AARC-1

Step-1: Given an input sentence in Eq. (1), initialization of the sentence matrix employing pre-trained embedding as per Eq. (5);
Step-2: for i in [1, M] do
Step-3: employ the BLSTM to extract its long-range richer context information according to Eq. (12);
Step-4: \( C^{l^2} \rightarrow \) capture all possible patterns by Eq. (14);
Step-5: \( A^{w_p} \rightarrow \) calculate attention vector using Eq. (18);
Step-6: obtain a sentence representation \( R^7 \) utilizing Eq. (20);
Step-7: end for
Step-8: Concatenate the \( S^T \rightarrow M \) sentence representation in the form of final featured matrix.
Step-9: \( Z^7 \rightarrow \) final output to similarity layer (SM\(^7\)), calculate distance using Eq. (21).
Step-10: Train all network parameters and update their weights by using the loss function with Adam-optimizer.

Algorithm: AARC-2

Step-1: Given an input sentence in Eq. (1), initialization of the sentence matrix employing pre-trained embedding as per Eq. (5);
Step-2: for i in [1, N] do
Step-3: employ the single directional LSTM to encode the context information \( L^7 \) by using Eq. (12);
Step-4: employ the CNN to encode significant semantic information \( C^N \) according to Eq. (15);
Step-5: concatenate the both \( M^7 \leftarrow L^7 + C^N \) in the form of one featured matrix.
Step-6: \( C^{l^2} \rightarrow \) capture all possible patterns as seen by Eq. (14);
Step-7: \( A^{w_p} \rightarrow \) calculate attention vector by Eq. (18);
Step-8: obtain a sentence representation \( R^7 \) using Eq. (20);
Step-9: end for
Step-10: Concatenate the \( S^k \rightarrow N \) sentence representation in the form of final featured matrix.
Step-11: \( Z^7 \rightarrow \) final output to similarity layer (SM\(^7\)), calculate distance using Eq. (21).
Step-12: Train all network parameters and update their weights by using the loss function with Adam-optimizer.

4. Experimental Results and Discussions

The proposed model has been executed for the above-mentioned purpose. Subsequently, the performance was evaluated on two benchmark datasets for sentence similarity
and compared with other methodologies. This section can be described as follows: the first sub-section illustrates pre-trained embedding, while the next one discusses datasets used in evaluating the model. This is followed by a comparison with other models. Finally, an experimental setup is discussed in detail.

4.1 Pre-Trained Embedding

In this section, available pre-trained word embedding setups are considered. The performance study of both the word2vec and the GloVe embedding methods has shown that they have a minor difference [37]. Here, GloVe is preferred for all of the datasets in this research. The dimension is set at 300 for every word vector. Absent words inside the pre-trained embedding vectors are initialized randomly. Word vectors are configured respectively during the training process.

- The word2vec method is trained on the Google News dataset (100 billion tokens) [31]. The model contains 300-dimensional vectors for 3 million words and phrases and was trained using continuous bag-of-words architecture.
- A GloVe is a 300-dimensional word embedding model learned on aggregated global word co-occurrence statistics from Common Crawl (840 billion tokens) [32].

4.2 Datasets

The datasets are concisely described as follows:

SICK stands for the Sentences Involving Compositional Knowledge dataset [14]. This data was taken from the 2014 SemEval competition. It is a computational semantic evaluation system prepared by the Special Interest Group on the Lexicon of the Association for Computational Linguistics. Its data contains 9840 sentence pairs with a 5154/4686 train/test split. The sentences are derived from an existing image and video description datasets. Every sentence pair is annotated with a relatedness label value $\in [1, 5]$, with a higher value indicating that two sentences are more closely similar.

MSRP stands for the Microsoft Research Paraphrase (MSRP) corpus [15]. This data was taken from news sources, which consists of 5,801 pairs of sentences, with 3,868 for training and the remaining 1,933 for testing. Every sentence pair is annotated with a binary value $[0, 1]$. It indicates whether the two sentences are paraphrases, and to assure binary classification, the 0 value indicates non-phrasing while 1 shows phrasing.

4.3 Comparison Systems

The experimental results of the new model are compared with different baseline methods for sentence embedding and similarity tasks.

- Manhattan (Ma) LSTM [21].
  The model belongs to two main network architectures LSTMa and LSTMb, where it shows that given enough data, a simple configuration of the LSTM enables it to train on sentence pairs.
- Tree-Structured LSTM [26].
  Tree-structured network topologies compose their architecture in the form an input vector and, the hidden states contain an arbitrary number of child units.
- CNN [25].
  Almost similar to the proposed model, it also has a two-tier architecture framework: ARC-I and ARC-II. This model uses a layer-by-layer composition and pooling to combine a set of word vectors in the form of sentence-level vector.
- Shallow CNN [27].
  The simplest form of CNN, it consists of only one feature layer, i.e. it is without any fully connected layer network. It proves that a fully connected layer is not vital for learning sentence vectors representation.

4.4 Experimental Setup

A detailed study of the sentence-classification task provides a guide to setting up CNN architecture and its hyper-parameters for practitioners [37]. The same mechanism is implemented to choose hyper-parameters for the model under discussion. Four hyper-parameters set for both datasets are listed in Table 1. The second item is a bidirectional LSTM layer with 128 cells. The hidden values go to a fully connected layer after the LSTM layer and subsequently, the values are passed towards the WPA to retain the attention vectors. The other hyper-parameters used are shown in Table 2. The average and maximum sentence length for both tasks are set from 15 to 100, correspondingly. According to the empirical analysis, it was found that the dropout helps us to improve the performance of the model, especially if the model is complex. It has been proved to be an efficient regularizer, while a high dropout rate leads to worse performance results. Moreover, due to an increased number of layers and modules, our model is computationally expensive. The proposed model is implemented in TensorFlow and Keras. All
of the experimental studies are conducted using an Intel Iris Pro 1536 GPU on Mac PC with core i7, CPU of 2.0 GHz and 8GB RAM. Experiments have been performed on different parameter values for LSTM cell units including dropout probability. Optimization is accomplished using the standard Adam optimizer.

4.5 Results

The test results of our model on a SICK dataset for semantic relatedness task-1 are shown in Fig. 4, where BLSTM and CNN-LSTM are named as ARC-1 and ARC-2, respectively, both utilized the WPA mechanism. All of the experiments have been conducted on pre-trained embedding systems as an input to BLSTM and LSTM-CNN. The Pearson ($r$) rank correlation coefficient and mean square error (MSE) are used as a measure for evaluating the models, where $r$ represents a standard correlation value used to measure the overall performance of the model. The MSE is an appropriate way to compute the average error rate, where smaller MSE values means that the predictive model has excellent accuracy. To this end, the rate of change can be evaluated in the model. When tested on the task-1 dataset, it gave the highest correlation value. For task-1, the results are summarized in Table 3, and were taken from the original field work like SemEval-Task-2014, mean vector, Tree-Structured LSTM, sequential LSTMs, and CNN to evaluate the model.

The six best outcomes of SemEval on task-1 are presented at the bottom of Fig. 5. The results of the SemEval on SICK dataset were released in 2014, and it can be clearly seen that the results of ECNU[39], Illinois-LH[41], The Meaning Factory[40], UNAL-NLP[42], Semanti-KLUE[43], CECLALL[44], DT-RNNA and SDT-RNN[45] are very close to each other. These models relied upon engineered features, which consisted of WordNet, and an additional corpus. In contrast, our approach does not need lexical resources and extensive manual features; it only requires pre-trained embedding and the hybrid structure of LSTM-CNN. The Pearson’s $r$ and MSE values of the proposed model are 0.0802, and 0.1729, respectively, which are higher than the best outcomes of SemEval task-1.

Figure 5 also demonstrates the comparison to mean vectors consisting of two models, DT-RNN and SDT-RNN. As discussed earlier, that the RNN is not effective for the handling long-term dependencies. The SDT-RNN is an upper-level variation of DT-RNN. It only computes the compositional vector representations as a mean of the representations of the constituent words. However, our model only calculates the attention vector instead of using the compositional vector. The models developed model here have Pearson’s $r$ value of 0.1197 and MSE of 0.233, comparatively better than the mean vector methods. The Dependency Tree-LSTM and Constituency Tree-LSTM models were trained on 150K and 319K labeled links, respectively. It relies on a dependency parser, while our approach does not require any syntactic parsing trees, and does not use POS tags or PARAGRAM embeddings. It only utilizes feature extractors like convolution filters and LSTM. The results of Tree-LSTM are compared with the proposed model; the Pearson’s correlation metrics and MSE are best around 0.0405 and 0.1011, approximately. Furthermore, another
work on sequential LSTM, which is trained on the spans related to labeled nodes, is listed in Fig. 5. All four results are lower than those of the proposed model. Their best results compared with our proposed model, Pearson’s $r$ is greater i.e., 0.0514. The suggested method achieved remarkable performance in terms of ‘$r$’ and MSE values compared with the baseline methods Manhattan (Ma) LSTM and Tree-Structured LSTM.

The Tree-LSTM method generalizes the order-sensitive LSTM model to complex tree-structured network topologies, where every sentence is converted into a parse tree. Thus, the training time is extensive. The MaLSTM typically needs a large corpus for training to achieve good outcomes due to the large number of parameters. Since our technique relies on pre-trained word vectors and has a simple hierarchical structure, our new model does not require any large set of datasets, but it also consists of numerous parameters and implicitly performs the semantic composition of an input. The comparison graph for task-1 as illustrated in Fig. 4 shows the performance of different methods. In an evaluation of the proposed model in comparison to the rest of the best existing systems, the formerly mentioned method surprisingly performs well with regard to Pearson’s $r$ and MSE values, which are 0.0259 and 0.0765, respectively.

Relatedness scores [21], [26] are also measures, used to analyze the performance of systems. To compare the relatedness, three test-set samples have been collected from Tree-LSTM and MaLSTM, as shown in Table 4. It was observed that the relatedness score of our proposed model is better than that of the baseline methods depicted in Fig. 6. The x-axis represents the baseline models compared with the proposed model, and the y-axis represents the relatedness label.

As the performance is also analyzed by conducting experiments on different datasets, we have conducted experiments on the MSRP corpus. The testing results on an MSRP used for paraphrase identification task-2 are shown in Fig. 7, where both the ARC-1 and the ARC-2 both architectures work together and perform well for the task-2. It can be seen that BLSTM+WPA with CNN+LSTM+WPA produced high accuracy. It also results in the highest accuracy value, when tested on the task-2. A comparison of the results with the existing state-of-art approaches is summarized in Table 5, where the gray highlighted results are based on deep learning methods. The proposed model does not acquire any additional sparse features, and its accuracy is 3.8% higher than that of Madnani et al. [46], which utilizes both pretraining learning and extra sparse features. The test results of Yin et al. [47], based on trained and pre-trained learning methods, are shown in Fig. 8.

A pre-trained technique has also been used in the suggested model, it has achieved better results and has surpassed traditional methods by 3.1%. Another model presented by Ji et al. [48], is an unsupervised learning method consisting of rich sparse features and has achieved better results close to the working model; nevertheless, our model slightly outperforms their best results. Moreover, our model also gave a superior performance than other recent neural network model approaches, e.g., Hu et al. [25], Socher et al. [49], He et al. [28] and Yao et al. [27]. These models do not need a large number of sparse features or unlabeled data for training as compared with our approach. The learn-

### Table 4 Most similar sentences data collected from Tree-LSTM [26]

| Ranking by Dependency Tree-LSTM | Tree-LSTM | MaLSTM | Proposed model |
|---------------------------------|-----------|--------|---------------|
| Model Tree                     |           |        |               |
| a woman is slicing potatoes     | 4.82      | 4.87   | 4.88          |
| a woman is cutting potatoes     | 4.70      | 4.38   | 4.50          |
| potatoes are being sliced by a woman | 4.39 | 3.51   | 4.10          |
| tofu is being sliced by a woman |          |        |               |
| a boy is waving at some young runners from the ocean | 3.79 | 3.13   | 3.80          |
| a group of men is playing with a ball on the beach | 3.37 | 3.48   | 3.60          |
| a young boy wearing a red swim suit is jumping out of a blue kiddies pool | 3.19 | 2.26   | 3.0           |
| the man is tossing a kid into the swimming pool that is near the ocean |          |        |               |
| two men are playing guitar      | 4.08      | 3.53   | 4.0           |
| the man is opening the guitar for donations and plays with the case | 4.01 | 3.20   | 4.10          |
| two men are dancing and singing in front of a crowd | 4.00 | 2.33   | 4.20          |

Fig. 6 Proposed model relatedness score test-set experimental results with baseline methods Tree-LSTM and MaLSTM.

Fig. 7 Results of ARC-1 and ARC-2 on MSRP dataset. Best accuracy results in each group are highlighted by the text.
Table 5 Proposed and recent state-of-the-art results on MSRP dataset for task-2. The gray highlighted results are deep learning methods and proposed work is highlighted as well.

| Models                        | Accuracy (%) |
|-------------------------------|--------------|
| Blake and Lapata (2012)       | 73.6         |
| Madanuk et al. (2012)         | 77.4         |
| Ji and Eisenstein (2013)      | 80.41        |
| Sucher et al. (2011)          | 76.1         |
| ARC-I                         | 69.6         |
| ARC-II                        | 69.9         |
| Hu et al. (2015)              |              |
| Pre-trained                   | 78.1         |
| Without pretrained            | 72.5         |
| Yin et al. (2015)             | 78.60        |
| (He et al. 2015)              |              |
| Pre-trained                   | 74.6         |
| Without pretrained            | 71.9         |
| Yao et al. (2017)             |              |
| BLSTM=WPA (ARC-1)             | 80.32        |
| CNN=LSTM=WPA (ARC-2)          | 80.61        |
| ARC-1+ARC-2                   | 81.20        |

The effectiveness of the weighted-pooling mechanism compared to existing pooling methods has been examined on the MSRP corpus for task-2. Moreover, all the modules and parameters of our models are the same except the pooling method. The experimental results of different strategies are depicted in Fig. 10, where 1-max, k-max and local-max pooling method results have better performance in comparison with average pooling. The weights setting in an average pooling scheme results in accuracy, which is worse than all of the other pooling approaches. Figure 10 also demonstrates that the results of 1-max and k-max pooling are too close to each other when tested with the MSRP dataset. It can be clearly seen that the local-max pooling strategy has a more satisfying performance than average pooling. The graph also depicts an evaluation of the proposed weighted pooling attention mechanism (our strategy), showing that it outperforms all of the other state-of-art pooling schemes. By this comparison, we have shown that the new proposed method achieves competitive performance due to its ability to efficiently capture the most significant information and
retain the order pattern of the given sequence.

It can be concluded from Fig. 5 and Fig. 8 that the WPA model persistently outperforms the other state-of-the-art systems in both tasks as compared with a simple RNN structure. Bidirectional-LSTM and hybrid LSTM-CNN helps our model to extract more comprehensive information for both short and long sentences and also increases the learning capability of the proposed model. The proposed model provides an effective WPA mechanism based on bidirectional LSTM and CNN networks. Experiments have proven its superior text semantic capturing ability compared to other existing attention mechanisms. In addition, the comparison with the recent state-of-the-art approaches manifests its competitive performance on all datasets and has established the efficiency of the proposed model to a recommended level.

5. Conclusion

In this research, a new neural network model termed weighted-pooling attention based on bidirectional long short-term memory and convolutional neural network for sentence similarity, has been presented. Our model has been successfully trained and a novel attention mechanism has been examined and tested. The attention mechanism helps to generate the sentence vector consisting of the most significant information. In addition, combining the LSTM-CNN has efficiently improved the model’s capability to extracting word-level, and sentence-level features with comprehensive contextual information. The developed model has also increased the learning capacity as well. The best value of Pearson’s rank correlation coefficient is 0.0259 and accuracy is enhanced up to 3%, thus obviously outperforming the existing baseline methods. Experiments have been performed on two distinct tasks and the results obtained exhibit the good learning capability of the proposed approach.

Addressing future prospects, this new approach can be applied to a variety of other natural language processing applications. Further, the incorporation of most well-known deep learning techniques such as auto-encoders, gated-recurrent, and convolutional generative adversarial neural networks can also be the subject of future research. Despite the fact that our model consists of a just one-dimensional CNN component, the only minor drawback in the suggested model is its huge computational complexity, which can be easily overcome by upcoming powerful computer systems. In the next study, endeavors will be undertaken to eradicate this drawback.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China under (No. U1936109, 61672127). First, we would like to thank the reviewers for their valuable comments and effort to improve the quality of the paper. The authors also wish to thank Dr. Cao Jingxiang for her useful suggestions, comments, and help during the editing of the manuscript.

References

[1] C. Kraus, “Plagiarism detection-state-of-the-art systems (2016) and evaluation methods,” arXiv Prepr. arXiv1603.03014, 2016.
[2] H. Li, “Learning to Rank for Information Retrieval and Natural Language Processing,” Synth. Lect. Hum. Lang. Technol., vol.4, no.1, pp.1–113, 2011.
[3] S. Minaee and Z. Liu, “Automatic question-answering using a deep similarity neural network,” 2017 IEEE Glob. Conf. Signal Inf. Process. Glob. 2017 - Proc., vol.2018–Janua, pp.923–927, 2018.
[4] M.C. Lee, J.W. Chang, and T.C. Hsieh, “A grammar-based semantic similarity algorithm for natural language sentences,” Sci. World J., vol.2014, 2014.
[5] Q.V. Le and T. Mikolov, “Distributed representations of sentences and documents,” International Conference on Machine Learning, pp.1188–1196, 2014.
[6] Q. Liu, Z.-H. Ling, H. Jiang, and Y. Hu, “Part-of-speech relevance weights for learning word embeddings,” arXiv Prepr. arXiv1603.07695, 2016.
[7] R. Collobert and J. Weston, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” Proc. 25th Int. Conf. Mach. Learn. - ICML ’08, pp.160–167, 2008.
[8] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, “Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval,” pp.1–25, 2015.
[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol.86, no.11, pp.2278–2323, 1998.
[10] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” arXiv Prepr. arXiv1404.2188, 2014.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in Neural Information Processing Systems, pp.5998–6008, 2017.
[12] I. Jolliffe, “Principal component analysis,” International Encyclopedia of Statistical Science, pp.1094–1096, Springer, 2011.
[13] J. Bromley, J.W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Suckling, and R. Shah, “Signature Verification Using a ‘Siamese’ Time Delay Neural Network,” Int. J. Pattern Recognit. Artif. Intell., vol.7, no.04, pp.669–688, 1993.
[14] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, “SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment,” Proc. 8th Int. Work. Semant. Eval. (SemEval 2014), no.1, pp.1–8, 2014.
[15] W.B. Dolan and C. Brockett, “Automatically constructing a corpus of sentential paraphrases,” Proc. Third Int. Work. Paraphrasing, pp.9–16, 2005.
[16] R. Mihalcea, C. Corley, and C. Strapparava, “Corpus-based and knowledge-based measures of text semantic similarity,” Proc. 21st Natl. Conf. Artif. Intell., vol.1, pp.775–780, 2006.
[17] Y. Wang, H. Huang, C. Feng, Q. Zhou, J. Gu, and X. Gao, “CSE: Conceptual sentence embeddings based on attention model,” Proc. ACL, vol.1, pp.505–515, 2016.
[18] P. Huang, N.M.A. Urbana, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” 22nd ACM Int. Conf. Inf. Knowl. Manag., pp.2333–2338, 2013.
[19] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval,” Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag. - CIKM ’14, pp.101–110, 2014.
[20] Z. Gan, Y. Pu, R. Henao, C. Li, X. He, and L. Carin, “Learn-
ing generic sentence representations using convolutional neural networks,” arXiv Prepr. arXiv1611.07897, 2016.

[21] J. Mueller, “Siamese recurrent architectures for learning sentence similarity,” Proc. 30th Conf. Artif. Intell. (AAAI 2016), no.2012, pp.2786–2792, 2016.

[22] T. Kim, J. Choi, D. Edmiston, S. Bae, and S. Lee, “Dynamic compositionality in recursive neural networks with structure-aware tag representations,” arXiv Prepr. arXiv1809.02286, 2018.

[23] A. Sanborn and J. Skryzalin, “Deep learning for semantic similarity - CS224d,” CS224d Deep Learn. Nat. Lang. Process. Stanford, CA, USA Stanford Univ., 2015.

[24] H. He and J. Lin, “Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement,” Proc. 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol., pp.937–948, 2016.

[25] B. Hu, Z. Lu, H. Li, and Q. Chen, “Convolutional neural network architectures for matching natural language sentences,” Advances in Neural Information Processing Systems, pp.2042–2050, 2015.

[26] K.S. Tai, R. Socher, and C.D. Manning, “Improved Semantic Representations From Tree-Structured Long-Short-Term Memory Networks,” Proc. 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Conf. Nat. Lang. Process., pp.1556–1566, 2015.

[27] H. Yao, H. Liu, and P. Zhang, “A novel sentence similarity model with word embedding based on convolutional neural network,” Concurr. Comput., no.Jan., 2018.

[28] H. He, K. Gimpel, and J. Lin, “Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks,” Proc. 2015 Conf. Empir. Methods Nat. Lang. Process. (EMNLP 2015), no.1, pp.1576–1586, 2015.

[29] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference on Learning Representations (ICLR), vol.5, 2014.

[30] R. Johnson and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” arXiv Prepr. arXiv1412.1058, 2014.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv Prepr. arXiv1301.3781, 2013.

[32] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1532–1543, 2014.

[33] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, “Advances in pre-training distributed word representations,” arXiv Prepr. arXiv1712.09405, 2017.

[34] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol.9, no.8, pp.1735–1780, 1997.

[35] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., no.3, pp.6645–6649, 2013.

[36] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol.521, no.7553, pp.436–444, 2015.

[37] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guides to) convolutional neural networks for sentence classification,” arXiv Prepr. arXiv1510.03820, 2015.

[38] F. Tao and G. Liu, “Advanced LSTM: A Study about Better Time Dependency Modeling in Emotion Recognition,” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.2906–2910, 2017.

[39] J. Zhao, T.T. Zhu, and M. Lan, “ECNU: One Stone Two Birds: Ensemble of Heterogeneous Measures for Semantic Relatedness and Textual Entailment,” Proc. 8th International Workshop on Semantic Evaluation (SemEval 2014), no.SemEval, pp.271–277, 2014.

[40] J. Bjerva, J. Bos, R. Van Der Goot, and M. Nissim, “The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity,” SemEval, no.SemEval, pp.642–646, 2014.

[41] A. Lai and J. Hockenmaier, “Illinois-LH: A Denotational and Distributional Approach to Semantics,” Proc. 8th Int. Work. Semant. Eval., no.SemEval, pp.329–334, 2014.

[42] S. Jimenez, G. Dueñas, J. Baquero, A. Gelbukh, G. Due, C. Universitaria, and A. Mendiz, “UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment,” Proc. 8th Int. Work. Semant. Eval. (SemEval 2014), no.1, pp.732–742, 2014.

[43] T. Proisl, S. Evert, P. Greiner, B. Kabashi, and F.E. Fau, “SemanticKLUE: Robust semantic similarity at multiple levels using maximum weight matching,” Proc. 8th Int. Work. Semant. Eval. (SemEval 2014), no.SemEval, pp.532–540, 2014.

[44] Y. Bestgen, “CECL: a New Baseline and a Non-Compositional Approach for the Sick Benchmark,” Proc. 8th Int. Work. Semant. Eval. (SemEval 2014), pp.160–165, 2014.

[45] R. Socher, A. Karpathy, Q.V. Le, C.D. Manning, and A.Y. Ng, “Grounded Compositional Semantics for Finding and Describing Images with Sentences,” Trans. Assoc. Comput. Linguist., vol.2, no.0, pp.207–218, 2014.

[46] N. Madnani, J. Tetreault, and M. Chodorow, “Re-examining machine translation metrics for paraphrase identification,” Proc. 2012 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol., pp.182–190, 2012.

[47] W. Yin and H. Schütze, “Convolutional neural network for paraphrase identification,” Proc. 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.901–911, 2015.

[48] Y. Ji and J. Eisenstein, “Discriminative improvements to distributional sentence similarity,” Proc. 2013 Conference on Empirical Methods in Natural Language Processing, pp.891–896, 2013.

[49] R. Socher, E.H. Huang, J. Pennington, A.Y. Ng, and C.D. Manning, “Dynamic pooling and unfolding recursive autoencoders for paraphrase detection,” Advances in Neural Information Processing Systems 24, pp.1–9, 2011.

[50] L. van der Maaten, and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol.1, pp.1–48, 2008.

Degen Huang received the B.S. degree in Computer Science from Fuzhou University, China, in 1986, and the M.S. and Ph.D. degrees in computer science from the Dalian University of Technology, China, in 1988 and 2004, respectively. He is currently a Professor with the School of Computer Science, Dalian University of Technology. His research interests include natural language processing, machine learning, and machine translation. He is now a senior member of CCF, CIPS, ACM, CAAI and an associate editor of Int. J. Advanced Intelligence.

Anil Ahmed received the B.S. degree in Computer Science from Mirpur University of Science & Technology, Mirpur, Pakistan in 2012. Currently, he is pursuing master’s degree in Computer Science and Technology from Dalian University of Technology, Dalian, P. R. China. His research interests include natural language processing, machine learning, Deep Learning, and machine translation.
Syed Yasser Arafat received the MSCS degree from International Islamic University (IIU), Islamabad in 2007. He is currently working as an Assistant Professor in Department of Computer Science and Information Technology (CS&IT), Mirpur University of Science and Technology (MUST). He has more than 14 years of teaching experience at various national universities. His research interests include NLP, Computer-Vision, Deep Learning, and Robotics. He is also pursuing his PhD. degree in Department of Computer Science, UET-Taxila, with research on Outdoor Urdu-Text Detection and Recognition.

Khawaja Iftekhar Rashid received the B.Sc. and MCS degree from University of Punjab, Lahore, Pakistan and University of Azad Jammu and Kashmir, Muzaffarabad, Pakistan in 2012 and 2014, respectively. Currently pursuing his master’s degree in Computer Science and Technology from Dalian University of Technology, Dalian. His research interests include Vehicular Networks, Network Information Theory, Wireless Sensor Networks, Autonomous Vehicles and Behavior Recognition.

Qasim Abbas a Pakistani national from Punjab province, had Bachelor of Sciences in 2009 and Masters in Physics in 2011 from University of Gujrat (UOG), Gujrat, Pakistan. He has been a lecturer in Physics Department of UOG from 2011 to 2014. He is currently continuing PhD in Microelectronics and Solid-State Electronics from School of Microelectronics and Physics, Dalian University of Technology, Dalian, China.

Fuji Ren received his Ph.D. degree in 1991 from the Faculty of Engineering, Hokkaido University, Japan. His current research interests include Natural Language Processing, Artificial Intelligence, Affective Computing, Emotional Robot. He is an academician of The Engineering Academy of Japan and EU Academy of Sciences. He is a senior member of IEEE, Editor-in-Chief of International Journal of Advanced Intelligence, a vice president of CAAI, and a fellow of The Japan Federation of Engineering Societies, a fellow of IEICE. He is the President of International Advanced Information Institute, Japan.