Research Article

A Deep Paraphrase Identification Model Interacting Semantics with Syntax

Leilei Kong , Zhongyuan Han , Yong Han, and Haoliang Qi

School of Electronic Information Engineering, Foshan University, Foshan 528225, China

Correspondence should be addressed to Zhongyuan Han; hanzhongyuan@gmail.com

Received 13 July 2020; Revised 25 September 2020; Accepted 4 October 2020; Published 30 October 2020

Academic Editor: Abd E. I.-Baset Hassanien

Copyright © 2020 Leilei Kong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Paraphrase identification is central to many natural language applications. Based on the insight that a successful paraphrase identification model needs to adequately capture the semantics of the language objects as well as their interactions, we present a deep paraphrase identification model interacting semantics with syntax (DPIM-ISS) for paraphrase identification. DPIM-ISS introduces the linguistic features manifested in syntactic features to produce more explicit structures and encodes the semantic representation of sentence on different syntactic structures by means of interacting semantics with syntax. Then, DPIM-ISS learns the paraphrase pattern from this representation interacting the semantics with syntax by exploiting a convolutional neural network with convolution-pooling structure. Experiments are conducted on the corpus of Microsoft Research Paraphrase (MSRP), PAN 2010 corpus, and PAN 2012 corpus for paraphrase plagiarism detection. The experimental results demonstrate that DPIM-ISS outperforms the classical word-matching approaches, the syntax-similarity approaches, the convolution neural network-based models, and some deep paraphrase identification models.

1. Introduction

The goal of paraphrase identification is to determine whether two texts have the same meaning [1]. It focuses on how best to model the semantics of sentences [2]. Paraphrase identification is one of the most basic problems in lots of applications of natural language processing, such as machine translation [3], question and answering [4], plagiarism detection [5,6], and document retrieval [7].

Although paraphrase identification is commonly defined in semantic terms [2], the early methods to paraphrase identification were usually based on the word (or word n-gram) matching or the vector similarity in the word space, without considering the semantics of words or sentences. The bag-of-words model [8], the n-gram model [9], the TFIDF [10] (term frequency and inverse document frequency) model, and so on were commonly applied to represent the text, and then some text similarity computing methods (such as edit distance, longest common substring, Jaccard coefficient, and cosine distance) were exploited to measure the degree of paraphrase between the two texts.

However, paraphrase is usually done by word replacement with synonyms/antonyms, syntactic modification, sentence reduction, combination, reorganization, word shuffling, concept generalization, and specificity to change the appearance of the original text while retaining the semantics of the source sentence [11], which makes the above methods difficult to further improve the performances using only word matching or vector similarity in the word space.

The syntactic feature-based methods, another way without considering the semantics, have also been used in paraphrase identification [11–13], especially in cross-language paraphrase identification [14]. These studies assume that similar texts have similar syntactic structures [12, 15]. That is, if two sentences describe the same thing, they are likely to have similar syntactic structures [16]. However, simply relying on the similarity of syntactic structures without regard to semantics cannot solve the problem of “the same semantics but different syntactic structures” [17].

In recent years, the models of paraphrase identification tend to transfer from the traditional model to deep model [18]. A variety of deep models have been introduced into the

research field of paraphrase identification [19–24]. These models utilized the distributed representation of text and focused on identifying the paraphrase through learning the matching structures and the matching degrees.

Except for the widely accepted distributed semantic representation in the deep paraphrase identification models, researchers also paid attention to the role of syntax in representing the text and computing the semantic similarity, and proposed some deep paraphrase identification models integrating the syntax [16, 25]. These studies determined the validity of syntactic features in deep paraphrase identification.

Goldberg presented that the linguistic features providing the more explicit general concepts can be very valuable [26]. Hu et al. proposed that a successful sentence-matching algorithm needs to capture not only the internal structures of sentences but also the rich patterns in their interactions [21]. We deem that the linguistic features manifested in syntactic features can produce more explicit structures for the representation of sentences and modeling the semantics on these syntactic features by means of the interaction of semantics with syntax can better represent the sentences and help to identify paraphrase.

Based on this, we propose a novel deep paraphrase identification model interacting semantics with syntax, denoted as DPIM-ISS. DPIM-ISS represents the sentences as the semantic vector on syntactic features and characterizes the syntactic role for the semantics of word or phrases by interacting semantic and syntactic information. Exploiting this representation, DPIM-ISS models the semantic representation on syntactic features explicitly and permits the model to learn the paraphrase pattern from the semantic on different linguistic features.

DPIM-ISS is evaluated on three datasets: MSRP (Microsoft Research Paraphrase) [27], PAN 2010 [6], and PAN 2012 [28]. The experimental results show that the proposed model outperforms the traditional word-matching approaches, the syntax-similarity approaches, the distributed-representations-of-sentences-based models, the CNN-based models, and a couple of deep models for paraphrase identification.

The contributions of this paper can be summarized as follows:

- (i) The idea of modeling the semantic representation of sentence on different syntactic structures by means of interacting semantics with syntax
- (ii) A new application of deep architecture, namely DPIM-ISS, to exploit the sentence representation interacting semantics with syntax for paraphrase identification
- (iii) Experiments on three datasets (i.e., MSRP, PAN 2010, and PAN 2012) to show the benefits of our model

The following sections are organized as follows: Section 2 analyzes the issues of paraphrase identification. Section 3 introduces the details of DPIM-ISS. The experimental results are reported in Section 4. Section 5 discusses the related work. Section 6 concludes our work.

2. Analysis of Paraphrase Identification

Taking the data of MSRP and PAN (the detailed statistics of the two datasets can be found in Section 4.1) as examples, we investigate the semantic similarity of the sentences from the aspects of lexical similarity and syntactic similarity to denote the paraphrase.

2.1. Paraphrase Sentences with High Lexical Similarity.

From the perspective of word matching, the sentences are more than likely being paraphrased if they use the same or similar words. We randomly selected 1000 pairs of paraphrase sentences and 1000 pairs of nonparaphrase sentences from the MSRP dataset and compared their lexical similarity using Jaccard coefficient, as Figure 1 shows.

Figure 1 reveals that when Jaccard coefficient is higher than 0.6, most of the sentence pairs are paraphrase sentences, while when Jaccard coefficient is lower than 0.25, most of the sentence pairs are nonparaphrase sentences.

Analyzing the examples of paraphrase sentences, we find that if the paraphrase sentences rewrite the source sentences by simple duplication, the syntactic structures of the two sentences are the same or similar, while if the paraphrase sentences rewrite the source sentences by text manipulation such as adjusting word orders or modifying the syntactic structures, the syntactic structures of the two sentences will therefore be different, but the words are still the same or similar. It shows that the word matching is still valuable in the paraphrase identification task. When Jaccard coefficients are between 0.25 and 0.6, it is difficult to distinguish paraphrase or nonparaphrase.

2.2. Paraphrase Sentences with the Same (Similar) Syntactic Structures but Different Words.

From the view of the syntactic structure, some paraphrase sentences have the same or similar syntactic structures but different words. Figure 2 gives a pair of paraphrase sentence from PAN 2012 with low lexical similarity but high syntactic similarity.

Figure 2 exemplifies a lexical paraphrase, where underlined words are replaced with synonyms, and short phrases or words are inserted to change the appearance of the text. Although much of the text is changed, paraphrasing retains the semantics of the source. It is a common type of case in paraphrase identification. The higher the degree of paraphrase, the more difficult to identify paraphrase only by word matching.

If the word matching is not considered and only the syntactic features are exploited, the pairs of such paraphrase sentences are more similar on syntactic structures. Figure 3 compares the Jaccard coefficients of syntactic features computed from 1000 pairs of paraphrase and nonparaphrase sentences randomly selected from the training dataset of MSRP. The X-axis records the Jaccard coefficients, and the y-axis is the number of the samples.

The statistical information in Figure 3 shows that the number of paraphrase sentence pairs is significantly higher than that of nonparaphrase sentence pairs as the similarity of the syntactic feature sequence increases. For example, when

FIGURE 1: Comparisons of lexical similarity between paraphrase pairs and nonparaphrase pairs.

Source sentence: In the early 1800's doc bowles built the first hotel, a three story frame building. The community thrived and there was an influx of tourist traffic coming to drink and soak in the mineral waters. In the 1850's French lick was a key station in the "underground railway". The French lick springs resort and spa was built in the late 1800's.
Paraphrased sentence: In an French ^{insertion} 1800's tourist ^{substitution} bowles built a ^{substitution} first drink ^{substitution} , the ^{substitution} three hotel railway mineral ^{substitution} . The community thrived and there was the ^{substitution} influx in ^{synonym} doc ^{substitution} traffic coming in story ^{substitution} and soak in the building ^{substitution} waters. To ^{synonym} the 1850's French lick was the early ^{substitution} station in the "underground frame ^{synonym} ". The key ^{substitution} lick springs resort and spa was built of a late ^{substitution} 1800's.

FIGURE 2: The samples of paraphrase sentence pair: different words, the similar syntactic structure, and the same semantics.

FIGURE 3: Comparison of syntactic similarity between paraphrase pairs and nonparaphrase pairs using Jaccard coefficient.

the Jaccard coefficients of the sentence pairs are between 0.8 and 0.9, there are 137 pairs of paraphrase sentences and only 28 pairs of nonparaphrase sentences. Therefore, the similarity of the syntactic structure is useful to the task of paraphrase identification.

2.3. Nonparaphrase Sentences with Similar Words and Similar Syntax Structures. Figure 4 describes an example of nonparaphrase sentences with similar words (black part) and similar syntax structures (see the dependency parsing tree corresponding to two sentences). In this example, S_1 and S_2 share a large number of the same words. Without respect to the semantics, the two sentences will be recognized as paraphrase due to the high levels of word matching.

Similarly, S_1 and S_2 can be identified as paraphrase since they have basically the same syntactic structures. However, if we compare the semantics of the words defined on the dependency tree, we can find that the semantics of verb appeared and surrendered are completely different, which leads to the semantic difference between the two sentences.

2.4. Different Words and Different Syntax, but the Same Semantics. Figure 5 shows an example in MSRP corpus with different words and different syntactic structure, but the same semantics.

In Figure 5, there are few identical words between two paraphrase sentences and the syntactic structures are much more varied. However, if we map the semantics of words to the substructures expressed by the dependency tree of sentences and compare the semantics of words in the syntactic substructures, such as refused and denied on VBD, the semantic similarity of the two sentences can be found.

A sentence written in the natural language is not the simple collection of words, but the text with the syntactic structure under the grammar restriction.

There exist the corelationships between semantics and syntax: when we need to convey and express the message in a proper way, the semantics and syntax of the sentence will work together, which encourages us to interact syntax and semantics in paraphrase identification to boost the performance.

3. Deep Paraphrase Identification Model Interacting Semantics with Syntax

The architecture of the deep paraphrase identification model interacting semantics with syntax (DPIM-ISS) contains two components: the sentence representation interacting semantics with syntax and the extraction of the matching pattern based on convolutional neural network. In this section, we introduce DPIM-ISS in detail.

3.1. Overview of DPIM-ISS. Paraphrase identification is usually formalized as a binary classification task [29]: given two sentences (s_k, s_p), the paraphrase identification model M determines whether they roughly have the same meaning. We propose DPIM-ISS to learn M , as shown in Figure 6.

In the architecture of DPIM-ISS illustrated in Figure 6, the model contains the two main parts: (1) the sentence representation interacting semantics with syntax, and (2) the extraction of the paraphrase matching pattern based on convolutional neural network. In what follows, we describe these components in detail.

3.2. The Sentence Representation Interacting Semantics with Syntax. In recent years, the tensor has attracted much attention due to its ability to model the interaction between objects. For example, Socher et al. proposed a neural tensor network to model the interaction of two entities [30] and Qiu et al. modeled the interaction between the questions and answers using tensor in the task of community question

FIGURE 4: Nonparaphrase sentence pair-similar words with similar syntax structure. (a) S_1 : He appeared in federal court on Monday and was expected to be transferred to Houston in two weeks. (b) S_2 : He surrendered in Cleveland on Monday and was expected to be transferred to Houston in two weeks.

FIGURE 5: Paraphrase sentence pair-different words with different syntax structure. (a) S_1 : The judge also refused to postpone the trial date of Sept. 29. (b) S_2 : Obus also denied a defense motion to postpone the September trial date.

answering [31]. In the study of Yu et al., the idea of tensor was exploited to model the interaction between the semantic information and the structural information [32]. The motivations of these methods are all to use tensor as the tool to capture the interaction between different features. Inspired by these studies, DPIM-ISS uses tensor to interact the semantics and syntactic structures to model the sentence representation. Figure 7 gives a detailed example.

Given a sentence $s_k = \{w_1^{(k)}, \dots, w_i^{(k)}, \dots, w_n^{(k)}\}$, where $w_i^{(k)}$ is the i -th word of s_k , let $e_{w_i}^{(k)}$ denote the semantic feature vector of $w_i^{(k)}$ represented as word embedding and $g_{w_i}^{(k)}$ be the syntax feature vector of $w_i^{(k)}$ that provides the syntax role of $w_i^{(k)}$. DPIM-ISS uses the tensor product $\otimes$ of $e_{w_i}^{(k)}$ and $g_{w_i}^{(k)}$ to project the structure of interacting semantics with syntax for the word $w_i^{(k)}$, represented by the notation $x_{w_i}^{(k)}$:

FIGURE 6: The architecture of the deep paraphrase identification model interacting semantics with syntax (DPIM-ISS).

FIGURE 7: Interacting semantics with syntax.

$$x_{w_i}^{(k)} = e_{w_i}^{(k)} \otimes g_{w_i}^{(k)}. \quad (1)$$

Let $g_{w_i}^{(k)} = \{g_1, g_2, \dots, g_m, \dots, g_M\}$ denote an example of predefined syntax feature vector template of size M . Each $g_{(m)}$ represents a fixed syntax feature such as the subject (the syntactic component), the noun (the speech), and so on. Given a word $w_i^{(k)}$, $g_{w_i}^{(k)}$ is a binary vector to represent the categorical variables. The categorical values are mapped to all zero values except those syntactic features that $w_i^{(k)}$ has. We use the syntactic parsing to obtain the syntax feature $g_{w_i}^{(k)}$ for the k -th word $w_i^{(k)}$ in sentence s . For example, in $g_{w_i}^{(k)}$, $g_{(m)} = 1$ means that $w_i^{(k)}$ has the m -th syntactic feature; on the contrary, $g_{(m)} = 0$ indicates that $w_i^{(k)}$ does not act as the role of m -th syntactic feature.

Using the semantic feature vector $e_{w_i}^{(k)}$ and the syntactic feature vector $g_{w_i}^{(k)}$, we then generate the word embedding representation $x_{w_i}^{(k)}$ interacting the semantics with syntax using equation (1). Each $x_{w_i}^{(k)}$ is a two-dimensional matrix, shown on the left of Figure 7. Furthermore, if each word in the sentence s_k is represented using the word embedding interacting the semantics with syntax, then we can get a

three-dimensional matrix M_{s_k} to represent the sentence s_k , shown on the right of Figure 7.

M_{s_k} consists of three dimensions: each word w in sentence s , the semantic feature vector e , and the syntactic feature vector g . DPIM-ISS captures the interactions between semantic features and syntactic features using tensor product, depicts the semantics of words on syntactic roles and decomposes the sentence into the syntactic subsections with semantics.

In order to obtain the expression of a sentence, we sum the word embedding interacting semantics with syntax to map M_{s_k} into a two-dimensional space of semantic and syntactic dimensions, shown in equation (2). Then, we obtain the representation $T^{(k)}$ of the sentence s_k , called the sentence representation interacting semantics with syntax in this paper:

$$T^{(k)} = \sum_{w_i \in s_k} x_{w_i}^{(k)}. \quad (2)$$

Furthermore, given two sentences s_k and s_p , we represent the interaction between them as a vector $A_{k,p}$ as follows:

$$A_{k,p} = T^{(k)} - T^{(p)}. \quad (3)$$

Then, the feature vector $A_{k,p}$ is further fed to a convolutional neural network to extract the paraphrase matching pattern.

3.3. Extracting the Paraphrase Matching Pattern Based on Convolutional Neural Network. The convolutional neural networks have been applied to learn effective feature representations in some language tasks in recent years. In DPIM-ISS, we use the convolutional neural networks to extract the features of paraphrase matching. Then, the extracted features will be fed into a multilayer perceptron classifier to identify the paraphrase.

3.3.1. Convolutional Layer. We use wide one-dimensional convolution [33], which was proposed by Kalchbrenner et al., to define the convolution kernel to extract the features from $A_{k,p}$ for paraphrase identification. In DPIM-ISS, $A_{k,p}$ is the interacting representation between the two sentences, and it is an $m \times n$ matrix, where m is the number of syntactic features and n is the dimension of semantic features.

The convolution layer exploits the U convolution kernels of size $1 \times n$ and a convolution kernel contains two parameters: W and b , where $W = [w_1, \dots, w_n]$ is the feature weight vector of the convolution kernel and b is the bias of the convolution kernel. A convolution kernel performs the convolutional operation on the interaction matrix $A_{k,p}$ to get an $m \times 1$ vector V_u , which represents the expression of a semantic feature on a syntactic feature. V_u is defined as follows:

$$V_u = \text{Cov}(W, A_{k,p}) + b, \quad (4)$$

where $\text{Cov}(W, A_{k,p})$ denotes a convolution operation on $A_{k,p}$ using parameter W :

$$\text{Cov}(W, A_{k,p}) = \begin{bmatrix} \sum_{i=1}^n w_i * A_{1,i} & \dots \\ \vdots & \vdots \\ \sum_{i=1}^n w_i * A_{m,i} \end{bmatrix}. \quad (5)$$

The convolution operation explores U convolution kernels to produce a matrix $A'_{k,p} \in R^{U \times m \times 1}$, which is composed by $m \times 1$ vector $V^{(u)} = [v_1^{(u)}, v_2^{(u)}, \dots, v_m^{(u)}]^T$, where m is the number of syntactic features and U is the number of convolution kernels:

$$A'_{k,p} = [V^{(1)}, V^{(2)}, \dots, V^{(u)}, \dots, V^{(U)}]. \quad (6)$$

3.3.2. Max Pooling. The outputs from the convolutional layer are then passed to the pooling layer to extract the k top values from each dimension of $A'_{k,p}$ for reducing the number of the features. On each column of $A'_{k,p}$, we set the size of nonoverlapping pooling window to w . The k features with the highest value are extracted from the window, and the matrix $A''_{k,p} \in R^{(U \times m/w \times 1)}$ made up of k m/w vectors is generated as follows:

$$A''_{k,p} = [V''^{(1)}, V''^{(2)}, \dots, V''^{(u)}, \dots, V''^{(U)}], \quad (7)$$

where each $V''^{(u)}$ is defined as follows:

$$V''^{(u)} = \begin{bmatrix} \max(v_1^{(u)}, \dots, v_w^{(u)}) \\ \max(v_{w+1}^{(u)}, \dots, v_{w+w}^{(u)}) \\ \vdots \\ \max(v_{\lfloor m/w \rfloor * w + 1}^{(u)}, \dots, v_m^{(u)}) \end{bmatrix}. \quad (8)$$

Then, the resulting features of $A''_{k,p}$ operated by max pooling are combined to form a $k \times m/w$ dimensional vector Z .

3.3.3. Further Enhancements. Madnani et al. proved that the machine translation (MT) metrics significantly boosted the performance of paraphrase identification [6]. For each pair of sentences, we construct a vector L to indicate the lexical similarity using the METEOR automatic MT evaluation metric, including precision, recall, F1, Fmean, penalty, and METEOR score [34]. We refer to such vector as the lexical features and incorporate it into the proposed DPIM-ISS by appending it to the vector Z . We conducted several experiments both with and without these features, which are discussed below.

3.3.4. Identifying Paraphrase. We pass Z with L to a two-layer perceptron, shown in equation (9):

$$(p_0, p_1)^T = \delta_2(W_2 \delta_1(W_1 Z + b_1) + b_2), \quad (9)$$

where p_0 and p_1 indicate the identification results, W_i and b_i are the weight matrix and the bias of the i -th layer of the perceptron, respectively, and δ_i is the ReLU activation function [35], defined as follows:

$$f(x) = \max(0, x), \quad (10)$$

and δ_2 is the SoftMax function to output the value of p_k :

$$p_k = \frac{e^{a_k}}{e^{a_0} + e^{a_1}}, \quad k = 0, 1, \quad (11)$$

where a_k is the output value after ReLU activation function in the last layer.

3.3.5. Training the Model. During the training phase, parameters of DPIM-ISS are updated with respect to a cross-entropy loss between the predicted results and the ground truth, and the regulation technology is adopted to avoid the overfitting problem. The loss function is defined as follows:

$$\begin{aligned} C(W, b) = & -\frac{1}{N} \sum_{M_i} [y^{(i)} \log(p_1^{(i)}) + (1 - y^{(i)}) \log(p_0^{(i)})] \\ & + \frac{\lambda}{2N} \sum_{W \in \{W_1, W_2\}} W^2, \end{aligned} \quad (12)$$

where $y^{(i)}$ is the label of i -th training example, λ is the regularization coefficient, and W_1 and W_2 are the parameters of the two-layer perceptron.

To train the model, we use the backpropagation algorithm [36] with the Adam update rule [37]. The updating forms of parameters are as follows:

$$W_t = W_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \quad (13)$$

where t is the current timestep, W_t is the weights of t -th timestep, W_{t-1} is the weights of W in the last round of training, η is the learning rate, ε is a parameter, and $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates to control the direction of the gradient. We set $\eta = 1e-4$ and $\varepsilon = 1e-08$.

The whole sentence representation interacting semantics with syntax and the training process are detailed in Appendix A.

4. Experiments

4.1. Datasets. We conduct our experiments on three datasets: the Microsoft Research Paraphrase (MSRP) [27], the PAN 2010 [6], and the PAN 2012 [28]. MSRP is a classical dataset for paraphrase identification developed by Microsoft, and the latter two datasets are constructed using the datasets of 2010 and 2012 Uncovering Plagiarism, Authorship and Social Software Misuse shared task.

4.1.1. MSRP. The MSRP corpus is a well-known corpus for paraphrase identification. MSRP was created by mining the news articles on the web and then extracting the paraphrases sentences from 9,516,684 sentences in 32,408 news clusters by using a semiautomatic method. It contains 5,801 sentential pairs, which is split into 4,076 (2,753 paraphrase, 1,323 not) training and 1,725 (1,147 paraphrase, 578 not) test pairs.

4.1.2. PAN 2010. Madnani and Tetreault used the human-created plagiarism instances in the test collection from the PAN 2010 plagiarism detection competition to create the PAN 2010 paraphrase sentence corpus. They utilized the bag-of-words overlap and length ratios to generate the pairs of paraphrase sentences and selected the sentence pairs that had at least 4 words in common from the same document as the pairs of nonparaphrase sentence. Then, they sampled randomly from both the positive and negative instances to create a training set of 10,000 sentence pairs and a test set of 3,000 sentence pairs.

4.1.3. PAN 2012. We constructed the PAN 2012 paraphrase sentence pair dataset using the training and test data of PAN 2012 paraphrase plagiarism detection corpus. Let d_{plg} and d_{src} denote the plagiarized document and its source document, and (s, r) is a pair of plagiarism text annotated by PAN ($s \in d_{\text{plg}}, r \in d_{\text{src}}$). Let $s_i \in s$ be the sentence of s and $r_j \in d_{\text{src}}$ denote the sentence of d_{src} , and $T = \{y, (s_i, r_j)\}$ represent the training dataset. y and r_j are defined as follows:

$$\begin{cases} y = 1, & r_j = \arg \max_{r_j \in r} (\cos(s_i, r_j)), \\ y = 0, & r_j = \arg \max_{r_j \in d_{\text{src}}, r_j \notin r} (\cos(s_i, r_j)), \end{cases} \quad (14)$$

where $\cos(s_i, r_j)$ is the cosine similarity of s_i and r_j . Using the proposed method, we obtained 341,426, and 50,114 pairs of paraphrase sentences from the artificial-high-obfuscation subcorpus of PAN 2012 training and test corpus. Then 15,932 training pairs and 7,966 test pairs that the length ratios were more than 50% were randomly selected to generate our PAN 2012 paraphrase sentence pairs dataset.

The statistics of the three datasets are described in Table 1.

4.2. Experimental Setting

4.2.1. Baselines. We evaluate the effectiveness of our model with several baseline methods, including the traditional word-matching approaches, the syntax-similarity approaches, the distributed-representations-of-sentences-based models, and the CNN-based models. At the same time, we also select multiple deep paraphrase identification models as baselines. We give a detailed description of these baselines as follows:

(1) Word-Matching Approaches. We select four typical word-matching approaches as baselines.

Jaccard. The Jaccard method calculates the Jaccard coefficient of the two sentences first and selects those pairs whose Jaccard coefficients are greater than a threshold t as the paraphrase sentence pairs. In experimenting, we set t from 0.0 to 1.0 and let the incremental step length be 0.01. We selected the parameter t on the training corpus in terms of optimizing accuracy. Then, the corresponding t was applied on the test corpus. On the MSRP dataset, $t = 0.34$. On PAN 2010, $t = 0.24$. On PAN 2012, $t = 0.27$.

Cosine. Similar to the Jaccard method, the cosine method calculates the similarity of the two sentences using the cosine distance. Similar to the above Jaccard method, we set a threshold t to decide the paraphrase sentence pairs. On the MSRP dataset, $t = 0.28$. On PAN 2010, $t = 0.34$. On PAN 2012, $t = 0.20$.

METEOR. We applied the six METEOR evaluation metrics as the features to learn a classifier using the logic regression model (in DPIM-ISS, these lexical features are integrated into the extracted features that interact semantics with syntax). All parameters are obtained based on the training data to optimize F1.

(2) Syntax-Similarity-Based Approaches (Syntax-sim). For syntactic similarity, we referred to the method proposed in [11], denoted as Syntax-sim (Syntax-similarity). In Syntax-sim, we considered the text as the string of syntactical sequences derived from Stanford POS tagging¹ instead of using actual words and utilized the Jaccard coefficient to compute the similarity of syntactical sequences for further decision.

TABLE 1: The statistics of the datasets.

Datasets		Training data	Test data
MSRP	Number of sentence pairs	4076	1725
	Short ≤ 20 words	2.40%	2.78%
	Medium 20–50 words	86.83%	86.03%
	Length of sentence pairs Long > 50 words	10.77%	11.19%
	Max length	60	63
	Min length	14	12
	$< 3\%$	0.02%	0.00%
	3%–10%	0.02%	0.12%
	10%–30%	13.10%	13.86%
	Jaccard coefficient 30%–50%	42.93%	43.65%
PAN 2010	50%–80%	41.78%	40.12%
	$> 80\%$	2.13%	2.26%
	Number of sentence pairs	10000	3000
	Short ≤ 50 words	35.76%	35.80%
	Medium 50–200 words	63.79%	63.93%
	Length of sentence pairs Long > 200 words	0.45%	0.27%
	Max length	477	272
	Min length	3	5
	$< 3\%$	0.24%	0.20%
	3%–10%	3.22%	2.43%
PAN 2012	10%–30%	57.71%	57.90%
	30%–50%	18.27%	18.13%
	50%–80%	18.52%	19.47%
	$> 80\%$	2.04%	1.87%
	Number of sentence pairs	15932	7966
	Short ≤ 50 words	51.82%	51.42%
	Medium 50–200 words	46.94%	47.39%
	Max length Long > 200 words	1.24%	1.19%
	Max length	1833	1658
	Min length	22	22
	$< 3\%$	0.01%	0.01%
	3%–10%	3.25%	3.09%
	10%–30%	75.71%	75.77%
	30%–50%	18.25%	17.95%
	50%–80%	2.74%	3.09%
	$> 80\%$	0.05%	0.09%

(3) *Distributed-Representations-of-Sentences-Based Model (Paragraph Vector)*. In our DPIM-ISS, we focus on the distributed representation of sentences. Thus, we select a distributed-representations-of-sentences-based model, the paragraph vector, proposed in [38] as the baseline for comparison. Paragraph vector used an unsupervised algorithm to learn the sentence representations. We utilized the tools of gensim² to learn the sentence vector and applied the cosine distance to compute the similarity of the two sentences. The parameter settings are as follows: the size of context window is 5, the lowest word frequency is 5, the learning rate is 0.025, and the dimension of sentence vector is 300.

(4) *CNN-Based Models*. ARC-I DPIM-ISS exploits the convolutional neural network to extract the paraphrase patterns of the interacting sentence representation. We also select a CNN-based paraphrase identification model, the ARC-I [21], as the baseline. In the experiment, we reimplemented ARC-I due to no publicly available codes, using the network structure and parameter setting as described in the original paper. The word embedding used for ARC-I was as the same as DPIM-ISS (will

be described in 5.2.3). All parameters were obtained based on training data to optimize F1.

(5) *Other Deep Paraphrase Identification Models*. We also compared the performance of DPIM-ISS with eight state-of-the-art deep models for paraphrase identification, including DSSM [19], CDSSM [20, 39], MV-LSTM [24], ARC-II [21], MatchPyramid [1], Match-SRNN [23], MP-DOT [1], and uRAE [25]. For DSSM, CDSSM, MV-LSTM, and Match-SRNN, the reported experimental results are provided by [18]. The experimental results of ARC-II, MatchPyramid, MP-DOT, and uRAE come from [1, 21, 22], respectively.

Except for the experimental results having been reported in the existing literature, all the parameters of the baselines and the DPIM-ISS are tuned to optimize the evaluation metrics F1 score on the training corpus and the best parameter settings are used on the testing corpus.

4.2.2. *Evaluation Metrics*. Followed the previous research, the task of paraphrase identification is formalized as a

classification problem and the accuracy and F1 score are used as the evaluation metrics. Accuracy can be formalized as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN}, \quad (15)$$

where TP is true positive, TN means true negative, FP is false positive, and FN represents false negative.

The F1 score is the harmonic mean of precision and recall:

$$F1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}, \quad (16)$$

where the precision and recall are defined as follows:

$$\begin{aligned} \text{precision} &= \frac{TP}{TP + FP}, \\ \text{recall} &= \frac{TP}{TP + FN}. \end{aligned} \quad (17)$$

4.2.3. Word Embedding. Word embedding required in the DPIM-ISS model and ARC-I was all learned based on One Billion Word Benchmark Corpus (<http://www.statmt.org/lm-benchmark/>) that contains nearly one billion sentences with different English words. We chose CBOW which was provided by gensim [40, 41] as the learning model. The dimension of word embedding was set to 300, the size of context window was set to 5, the lowest word frequency was 5, and the learning rate was 0.0002.

4.2.4. Syntactic Features. We used Stanford’s parser (<https://nlp.stanford.edu/software/lex-parser.shtml>) to get the dependency tree of sentences. The results of parser described the syntactic relationship in a sentence by means of the part of speech and the interword dependency. In our experiment, we only preserved the part-of-speech tags and the word dependency tags. These markers were used as the syntactic features, and we simplified these tags in our experiment. For example, we simplified the tag `nmod:including` as `nmod`. Then, only 30 syntactic tags were preserved, shown in Table 2.

4.3. Experimental Results and Analysis. The experimental results are summarized in three parts. In Section 4.3.1, we compare DPIM-ISS to the traditional word-matching approaches, the syntax-similarity approaches, the distributed-representations-of-sentences deep models, and the CNN-based models. We compare the performances of DPIM-ISS with other deep models for paraphrase identification in Section 4.3.2. In Section 4.3.3, we analyze the performance of each substructure in our model.

4.3.1. Comparison with the Word-Matching Approaches, the Syntax-Similarity Approaches, and the Distributed-Representations-of-Sentences Deep Models. The main comparison results of our experiments on MSRP, PAN 2010, and PAN 2012 are summarized in Table 3.

First, we compare the performance of DPIM-ISS with word-matching approaches. We observe that the DPIM-ISS outperforms the Jaccard approach, the Cosine approach, and the METEOR approach on F1 score and accuracy. Comparing DPIM-ISS with METEOR, the experimental results show that DPIM-ISS performs better than the method using only lexical features.

In addition, on PAN 2010 and PAN 2012 datasets, the METEOR approach, which takes the synonym matching into account, is significantly higher than the baselines on accuracy and F1 score. This is closely related to the synonym replacement method used in the construction of PAN datasets.

Then, we analyze the performance of DPIM-ISS and syntax-similarity approaches. The experimental results show that the DPIM-ISS has a significant improvement over the syntax-similarity approach. We also note that the improvement on MSRP datasets is lower than that on PAN 2010 and PAN 2012 datasets. Similarly, compared DPIM-ISS with the method Sentence2Vector and ARC-I, we found that the performance improvements on MSRP are lower than those on PAN 2010 and PAN 2012. We conclude that the performance gap is attributed to the construction methods of the MARP dataset and PAN datasets.

For analyzing the differences on performance, we investigate the three datasets and find two main issues: (1) the syntactic structure on MSRP is more similar than those on PAN datasets. (2) Compared with MSRP, the use of words of PAN are significantly different.

Since the MSRP dataset was constructed using the corpus of topic-clustered news data, it does not adopt the deliberate obfuscation, which results in the small lexical differences but similar syntactic structures between the two sentences in MSRP. Therefore, DPIM-ISS does not get much more benefits than the traditional deep learning methods. For the two PAN datasets, the source sentences are paraphrased in order to avoid plagiarism detection. The vocabulary shows the significant variations, and the syntactic structure takes on the marked difference. By decomposing the sentence’s syntactic structure using the dependency tree, we obtain the key substructures of a sentence. The same substructures may be owned by the two sentences simultaneously (such as the predicate verb). Although these substructures present different appearance in terms of words, they may have similar semantics. DPIM-ISS uses the sentence expression interacting the semantics with syntax to obtain the semantic expression on the syntactic structures and learns the patterns of paraphrase in these semantic expressions using CNN. It pays attention to the different functions of semantic matching in different syntactic structures on paraphrase identification and solves the issues of the different syntactic structures as well as the different words to a certain extent.

4.3.2. Comparison with Other Deep Models for Paraphrase Identification. Based on the MSRP dataset, we compare the performance of DPIM-ISS with other main deep models for paraphrase identification. We choose the MSRP dataset

TABLE 2: Syntactic features.

No.	Feature	No.	Feature	No.	Feature	No.	Feature	No.	Feature
1	advcl	7	JJR	13	RB	19	dobj	25	nsubjpass
2	advmod	8	JJS	14	RBR	20	FW	26	nummod
3	ccomp	9	neg	15	RBS	21	iobj	27	VBG
4	CD	10	NN	16	root	22	JJ	28	VCN
5	csubj	11	NNP	17	VB	23	NNS	29	VBP
6	csubjpass	12	NNPS	18	VBD	24	nsubj	30	VBZ

TABLE 3: Performance comparisons with word-matching-based approaches, the syntax-similarity approaches, the text-semantic-representation-based deep models, and the CNN-based models.

		MSRP		PAN 2010		PAN 2012	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
Word-matching-based	Jaccard	72.06	81.53	86.26	85.86	53.53	69.73
	Cosine	70.89	81.69	85.23	84.87	65.12	67.45
	METEOR	73.10	81.06	89.50	88.90	82.11	80.70
Syntax-similarity-based	Syntax-sim	66.90	80.03	74.57	72.10	62.74	69.65
Text-semantic-representation-based	Paragraph vector	67.42	80.21	67.33	70.45	51.08	66.48
Deep models	ARC-I	69.60	80.27	50.01	66.68	50.14	66.39
Our model	DPIM-ISS	73.57	83.55	91.10	91.07	83.60	82.56

TABLE 4: Comparison with other deep models for paraphrase identification on the MSRP dataset.

Deep models	Accuracy	F1
DSSM	70.09	80.96
CDSSM	69.80	80.42
MV-LSTM	75.40	82.80
ARC-II	69.90	80.91
MatchPyramid	75.94	83.01
Match-SRNN	74.50	81.70
MP-DOT	75.94	83.01
uRAE	76.8	83.60
DPIM-ISS	73.57	83.55

since the results of various deep models for paraphrase identification can be obtained directly from the literature which proposed these models. The data listed in Table 4 come from the experimental results presented in the corresponding literature.

From Table 4, we can see that uRAE and DPIM-ISS, which are built based on the syntactic information, perform much better than the other baselines. Though the best performance of our model (83.55) is still slightly worse than uRAE on F1 score (83.6%) [22], uRAE relies heavily on pretraining on an external large dataset annotated with parse tree information to learn the representation of phrase features for each node in a parse tree. Compared with uRAE, DPIM-ISS only needs to parse the two sentences to be recognized for obtaining the syntactic structures without any additional pretraining.

4.3.3. Model Analysis. First, we analyze the influence of lexical features on DPIM-ISS. We remove the lexical features in DPIM-ISS and use the features captured by the convolutional neural network from the interacting sentence expression as the input of MLP directly to learn the classifier.

TABLE 5: The effect of lexical features on DPIM-ISS.

Model	MSRP		PAN 2010		PAN 2012	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
DPIM-ISS-L	70.50	81.84	86.77	87.74	68.40	72.53
DPIM-ISS	73.57	83.55	91.10	91.07	83.60	82.56

The model that removes the lexical features is denoted as DPIM-ISS-L. Table 5 lists the performance comparison between DPIM-ISS-L and DPIM-ISS.

The experimental results in Table 5 demonstrate that the lexical features help to improve the performance of paraphrase identification, especially on the PAN 2012 dataset. We conclude that METEOR evaluation measures take the synonym replacement into account, which is one of the main construction strategies of the PAN 2012 dataset. However, on the MSRP dataset, there are little changes in the use of words and the syntactic structures, so the additional lexical features do not lead to a significant improvement on MSRP than on PAN 2012.

For the number of syntactic feature parameters, we compared the performance of 30 syntactic features with 67 syntactic features. On MSRP training corpus, we got 0.7119 on accuracy and 0.8173 on F1 when we used 30 syntactic features (the syntactic features in Table 2). However, when we used 67 syntactic features (30 syntactic features in Table 2 added another 37 syntactic features), we got 0.6805 on accuracy and 0.8041 on F1. We also tried two commonly used dimensions of word embedding, 300 and 600, on MSRP training corpus. The accuracy got by the 300 dimensions word embedding was 0.7119 on accuracy and 0.8173 on F1, while the 600-dimensional word embedding achieved 0.6786 on accuracy and 0.7891 on F1. The above two experimental results show that too many features will

affect the classification performance on the size of network that we designed. To further improve the performance of DPIM-ISS, we can try to expand the network size or add the network layers to enhance the representation ability of DPIM-ISS.

5. Related Work

Early work on paraphrase identification usually relied on lexical, semantic, or syntactic similarity measures to identify paraphrases.

Lexical-based approaches used the bag-of-words representations without considering the semantics of the words, which inevitably led to the problem of “polysemy” and “synonymy” in paraphrase identification.

Some methods resorted to the knowledge base (such as WordNet) to measure the word semantic similarity for alleviating the restrictions of word matching-based paraphrase identification methods. For example, Mihalcea et al. utilized the WordNet-based measures to compute the word semantic similarity [8], Mohamed and Oussalah also presented to use the WordNet and Wikipedia to compute the word semantic similarity and named-entity semantic relatedness for paraphrase identification [42], Madnani et al. exploited the METEOR (based on WordNet) machine translation metrics as the features of classifiers to determine the paraphrase [6], and Islam et al. [43] and Bollegala et al. [44] computed semantic similarity using a corpus-based measure. The main advantage of knowledge base-based semantic approaches is that it can make full use of the prior knowledge of experts. However, the limitations of this kind of approaches mainly include the following: knowledge base needs the human maintenance and updating, the limitation of vocabulary coverage, and the lack of sufficient context information to determine the exact concepts.

On the other hand, researchers have noticed the role of syntactic features in paraphrase identification and presented some syntax-based methods. For example, Das and Smith believed that the paraphrase was related to the syntactic structure, and they used the part-of-speech tag and the syntactic dependence of words as the features to learn the classifier [2], Koroutchev et al. exploited the Lempel–Ziv algorithm to compare the syntactic and morphological features of the two texts to detect the text similarity [13], Elhadi and Al-Tobi utilized the part-of-speech sequence to represent text and detect plagiarism [12, 15], Pothast et al. employed n-grams of the syntactic structure sequence to detect the plagiarism in European languages [14], and Mohammad et al. extracted the POS tags as syntactic features of classifiers to identify the paraphrase for the Arabic language [45]. However, these methods could not work effectively when the syntactic structures changed greatly.

To avoid the disadvantages of single class of similarity measures, a different way to look to paraphrase identification is relying on the supervised learning to combine the lexical, syntactic, and semantic features to classify the sentence pair paraphrase or not [46].

In recent years, the distributed representation of words or text has made progress of the semantic representation. Manning pointed out that having a dense, multidimensional representation of similarity between all words was incredibly useful in natural language processing [47]. The distributed representation uses the vectors in contiguous semantic space to project the linguistic units, which makes the similarities of words can be calculated using the distances of word vectors. Thus, two sentences, represented as two vectors in the low-dimensional semantic space, can still have a high similarity even if they do not share any term [39].

Inspired by the success of the deep neural networks recently, the paraphrase identification has been innovated towards the deep paraphrase identification models, including the full-connected neural network-based models such as DSSMs (deep structured semantic models) [19], the CNN-based (convolution neural network) models such as CDSSMs (convolutional deep structured semantic models) [20, 39], ARC-I (Architecture-I) [21], ARC-II (Architecture-II) [21], MatchPyramid [1] and Match-SRNN (Match-special recurrent neural network) [23], the recurrent neural network-based (RNN) models such as MV-LSTM (MV-bidirectional long short-term memory) [24], CNN- and RNN-based models such as Deep-Paraphrase [48], and attention-based alignment models such as pt-DecAtt [49]. These methods focused on the distributed representation of text and identified the paraphrase through the learning of matching degrees and matching patterns, which reduces the dependence on the design of artificial features.

Researchers also introduced the features of syntactic structures into the framework of deep paraphrase identification models. For example, Socher et al. deemed that syntactic and semantic analysis was needed for paraphrase detection, and they presented to exploit recursive autoencoders (RAEs) and unfolding recursive autoencoder (uRAE) to encode the words, the multiword phrases, and the sentences in syntactic trees [25]. Zhou et al. followed the idea of Socher and used the weighted uRAE to encode the phrases and sentences embedding that obtained from parse trees [50]. Wang et al. proposed the DeepMatch_{Tree} to match the two short texts that relied on a tree-mining algorithm [16]. Based on the dependency tree, DeepMatch_{Tree} represented the two sentences as the binary matching models composed by the subtree pairs and utilized a deep neural network to learn the matching pattern. Considering the influence of syntactic structure on semantic computation, Liu et al. [51] exploited the syntactic feature for paraphrase identification. In their method, based on the syntactic tree, the TreeLSTM [52] was used to model the sentences and represent the semantic composition. Especially, they introduced the attention mechanism to extract the cross-sentence features. Xu et al. also made use of syntactic features to indicate the dependency relation between words [53]. They incorporated the lexical, syntactic, and sentential encodings for paraphrase identification. In their approach, integrating the syntactic features was verified to contribute to performance improvement. However, the high performance cannot be divorced from the large-scale pretrained model, such as

```

INPUT:  $S = \{(y_{kp}, (s_k, s_p))\}$ , iterations
OUTPUT: model
for  $(sk, sp)$  in  $S$ :
   $(e_{w_1}^{(k)}, e_{w_2}^{(k)}, \dots, e_{w_i}^{(k)}, \dots, e_{w_n}^{(k)}) \leftarrow \text{Embedding}(s_k)$ ,  $(e_{w_1}^{(p)}, e_{w_2}^{(p)}, \dots, e_{w_i}^{(p)}, \dots, e_{w_n}^{(p)}) \leftarrow \text{Embedding}(s_p)$ 
   $(g_{w_1}^{(k)}, g_{w_2}^{(k)}, \dots, g_{w_i}^{(k)}, \dots, g_{w_n}^{(k)}) \leftarrow \text{SyntaxParsing}(s_k)$ ,  $(g_{w_1}^{(p)}, g_{w_2}^{(p)}, \dots, g_{w_i}^{(p)}, \dots, g_{w_n}^{(p)}) \leftarrow \text{SyntaxParsing}(s_p)$ 
  for  $i$  in  $1..n$ :
     $x_{w_i}^{(k)} \leftarrow e_{w_i}^{(k)} \otimes g_{w_i}^{(k)}$ ,  $x_{w_i}^{(p)} \leftarrow e_{w_i}^{(p)} \otimes g_{w_i}^{(p)}$ 
     $T^{(k)} \leftarrow T^{(k)} + x_{w_i}^{(k)}$ ,  $T^{(p)} \leftarrow T^{(p)} + x_{w_i}^{(p)}$ 
   $A_{k,p} \leftarrow T^{(k)} - T^{(p)}$ 
   $z_{k,p} \leftarrow \text{ExtractingLexicalFea}(s_k, s_p)$ 
   $T \leftarrow \text{append}(y_{k,p}, A_{k,p}, z_{k,p})$ 
for iter in range(iterations):
  model  $\leftarrow \text{TrainingModel}(T)$ 
return model.

```

ALGORITHM 1: Training DPIM-ISS.

BERT (bidirectional encoder representations from transformers) [54].

The above approaches enjoyed the advantages of integrating the syntactic features in the paraphrase identification. They all exploited the dependency trees to obtain the local substructures of words or phrases on the syntactic structures at different granularities and learned the semantic representation of these substructures. In this regard, the ideas of this paper are the same as those of the existing work. The difference lies in the semantic representation and interaction on syntactic structures. DPIM-ISS is designed to interact the semantics and syntactic features for obtaining the semantic representation on syntactic structures. Furthermore, we exploit the explicit syntactic structure to model the semantic interaction on syntactic structures between two sentences. This allows us to learn the paraphrase pattern from the semantics on different linguistic features, which was not performed in the RAE, uRAE, weighted uRAE, and DeepMatch_{Tree}.

6. Conclusions

In this paper, we present the DPIM-ISS, a novel text deep paraphrase identification model interacting semantics with syntax. In DPIM-ISS, we introduce the syntactic information by capturing the syntactic structures and represent the semantics by means of the distributed representation method. Then, we exploit the tensor to interact the semantics and syntax for representing the sentences and use the convolutional neural network to extract the paraphrase patterns in text matching space. Experiments on MSRP, PAN 2010, and PAN 2012 corpus demonstrate that DPIM-ISS achieves comparable or better performance against the traditional word-matching approaches, the syntax-similarity approaches, the distributed-representations-of-sentences-based models, the CNN-based models, and some text deep paraphrase identification methods.

There is an important direction to improve the performance of DPIM-ISS. We note that the acquisition of syntactic features now mainly relies on the results of

syntactic parsing. The advantage of this kind of approach is to capture the explicit syntactic structures. However, we can try to another way of exploiting syntactic features, for example, to integrate the representation and the learning of the syntactic features into the network of DPIM-ISS directly. This should be one of our future work.

Appendix

A. Algorithm for Sentence Representation Interacting Semantics with Syntax and Training Process

Sentence representation interacting semantics with syntax and training process is presented in Algorithm 1.

Data Availability

The data of MSRP are available at <https://www.microsoft.com/en-us/download/details.aspx?id=52398>, the data of PAN 2010 are available at <http://bit.ly/mt-para>, and the data of PAN 2012 are available at <https://pan.webis.de/data.html#pan12-text-alignment>. The data used to support the findings of this study are also available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 61806075 and 61772177).

References

- [1] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 2793–2799, Phoenix, AZ, USA, February 2016.

- [2] D. Das and N. A. Smith, "Paraphrase identification as probabilistic quasi-synchronous recognition," in *Proceedings of the the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 468–476, Singapore, August 2009.
- [3] C. Callison-Burch, P. Koehn, and M. Osborne, "Improved statistical machine translation using paraphrases," in *Proceedings of the the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 17–24, New York, NY, USA, June 2006.
- [4] X. Xue, J. Jeon, and W. B. Croft, "Retrieval models for question and answer archives," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 475–482, Singapore, July 2006.
- [5] P. Clough, R. Gaizauskas, S. S. L. Piao, and Y. Wilks, "METER: MEasuring TExt Reuse," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 152–159, Philadelphia, PA, USA, July 2002.
- [6] N. Madnani, J. Tetreault, and M. Chodorow, "Re-examining mtranslation metrics for paraphrase identification," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 182–190, Montréal, Canada, June 2012.
- [7] H. Li and J. Xu, "Semantic matching in search," *Foundations and Trends in Information Retrieval*, vol. 7, no. 5, pp. 343–469, 2014.
- [8] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the 21st National Conference on Artificial Intelligence-Volume1*, AAAI Press, Boston, MA, USA, pp. 775–780, 2006.
- [9] Y. Zhang and J. Patrick, "Paraphrase identification by text canonicalization," in *Proceedings of the Australasian Language Technology Workshop*, pp. 160–166, Sydney, Australia, 2005.
- [10] W. Guo and M. Diab, "Modeling sentences in the latent space," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 864–872, Stroudsburg, PA, USA, 2012.
- [11] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding plagiarism linguistic patterns, textual features, and detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 133–149, 2012.
- [12] M. Elhadi and A. Al-Tobi, "Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures," in *Proceedings of the 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, pp. 679–684, Seoul, Korea, November 2009.
- [13] K. Koroutchev and M. Cebrián, "Detecting translations of the same text and data with common source," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 10, Article ID P10009, 2006.
- [14] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 45–62, 2011.
- [15] M. Elhadi and A. Al-Tobi, "Use of text syntactical structures in detection of document duplicates," in *Proceedings of the third IEEE International Conference on Digital Information Management (ICDIM)*, London, UK, November 2008.
- [16] M. Wang, Z. Lu, H. Li, and Q. Liu, "Syntax-based deep matching of short texts," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 1354–1361, Buenos Aires, Argentina, July 2015.
- [17] N. Chomsky, "The logical basis of linguistic theory," in *Proceedings of the 9th International Congress of Linguists*, Mouton and Co., Hague, Netherlands, pp. 914–978, 1964.
- [18] L. Pang, Y. Lan, J. Xu, J. Guo, S. Wan, and X. Cheng, "A survey on deep text matching," *Chinese Journal of Computers*, vol. 39, no. 126, pp. 985–1003, 2016.
- [19] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, pp. 2333–2338, San Francisco, CA, USA, October 2013.
- [20] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374, Seoul, Korea, April 2014.
- [21] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2042–2050, Montréal, Canada, December 2014.
- [22] W. Yin and H. Schütze, "MultiGranCNN: an architecture for general matching of text chunks on multiple levels of granularity," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pp. 63–73, Beijing, China, July 2015.
- [23] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Match-SRNN: modeling the recursive matching structure with spatial RNN," *Computers & Graphics*, vol. 28, no. 5, pp. 731–745, 2016.
- [24] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 2835–2841, Phoenix, AZ, USA, February 2016.
- [25] R. Socher, E. H. Huang, J. Pennin, A. Y. Ng, and C. D. Manning, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, pp. 801–809, Granada, Spain, December 2011.
- [26] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.
- [27] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing*, pp. 9–16, Jeju Island, Korea, January 2005.
- [28] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, "An evaluation framework for plagiarism detection," in *Proceedings of the 23rd International Conference on Computational Linguistics*, Association for Computational Linguistics, Beijing, China, pp. 997–1005, August 2010.
- [29] W. Yin and H. Schütze, "Convolutional neural network for paraphrase identification," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 901–911, Denver, CO, USA, May 2015.
- [30] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion,"

- in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 926–934, Lake Tahoe, USA, 2013.
- [31] X. Qiu and X. Huang, “Convolutional neural tensor network architecture for community-based question answering,” in *Proceedings of the International Conference on Artificial Intelligence*, pp. 1305–1311, Buenos Aires, Argentina, July 2015.
 - [32] M. Yu, M. R. Gormley, and M. Dredze, “Combining word embeddings and feature embeddings for fine-grained relation extraction,” in *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1374–1379, Denver, CO, USA, May 2015.
 - [33] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Baltimore, MA, USA, pp. 655–665, June 2014.
 - [34] S. Banerjee and A. Lavie, “METEOR: an automatic metric for MT evaluation with improved correlation with human judgments,” *The ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation And/or Summarization*, vol. 29, pp. 65–72, 2005.
 - [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.
 - [36] D. Williams and G. Hinton, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–538, 1986.
 - [37] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *Computer Science*, vol. 3, pp. 1–13, 2015.
 - [38] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” *Computer Science*, vol. 4, pp. 1188–1196, 2014.
 - [39] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “A latent semantic model with convolutional-pooling structure for information retrieval,” in *Proceedings of the ACM International Conference on Conference on Information and Knowledge Management*, pp. 101–110, Shanghai, China, November 2014.
 - [40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, AZ, USA, May 2013.
 - [41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their Compositionality,” in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013.
 - [42] M. Mohamed and M. Oussalah, “A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics,” *Language Resources and Evaluation*, vol. 54, no. 2, pp. 457–485, 2020.
 - [43] A. Islam and D. Inkpen, “Semantic text similarity using corpus-based word similarity and string similarity,” *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, pp. 1–25, 2008.
 - [44] D. Bollegala, Y. Matsuo, and M. Ishizuka, “A web search engine-based approach to measure semantic similarity between words,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 977–990, 2011.
 - [45] A. L. S. Mohammad, Z. Jaradat, A. L. A. Mahmoud et al., “Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features,” *Information Processing & Management*, vol. 53, no. 3, pp. 640–652, 2017.
 - [46] R. Ferreira, G. D. C. Cavalcanti, F. Freitas, R. D. Lins, S. J. Simske, and M. Riss, “Combining sentence similarities measures to identify paraphrases,” *Computer Speech & Language*, vol. 47, pp. 59–73, 2018.
 - [47] C. D. Manning, “Computational linguistics and deep learning,” *Computational Linguistics*, vol. 41, no. 4, pp. 699–705, 2015.
 - [48] B. Agarwal, H. Ramampiaro, H. Langseth, and M. Ruocco, “A deep network model for paraphrase detection in short text messages,” *Information Processing & Management*, vol. 54, no. 6, pp. 922–937, 2018.
 - [49] G. S. Tomar, T. Duque, O. Täckström et al., “Neural paraphrase identification of questions with noisy pretraining,” in *Proceedings of the First Workshop on Subword and Character Level Models in NLP, Association for Computational Linguistics 2017*, pp. 142–147, Copenhagen, Denmark, September 2017.
 - [50] J. Zhou, G. Liu, and H. Sun, “Paraphrase identification based on weighted URAE, unit similarity and context correlation feature,” in *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 41–53, Hohhot, China, August 2018.
 - [51] M. Liu, Y. Zhang, and Y. Chen, “A neural paraphrase identification model based on syntactic structure,” *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 56, no. 1, pp. 45–52, 2020.
 - [52] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured Long Short-Term Memory networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1556–1566, Beijing, China, 2015.
 - [53] S. Xu, X. Shen, F. Fukumoto, J. Li, Y. Suzuki, and H. Nishizaki, “Paraphrase identification with lexical, syntactic and sentential encodings,” *Applied Sciences*, vol. 10, no. 12, p. 4144, 2020.
 - [54] J. Devlin, M. W. Chang, K. Lee et al., “BERT: pre-training of deep bidirectional Transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, Minneapolis, MN, USA, 2019.