MSF-Net: Multi-level Semantic Feature Network Extractor for Paraphrase Identification

Wenrui Xue, Yujun Zhang, Xinyu Wang, and Shuting Ge

***Abstract*—To address inaccurate semantic representation and challenges in interpreting rare words within deep learning-based paraphrase identification tasks, this paper introduces a multi-level semantic feature network extractor (MSF-Net). The MSF-Net model represents an end-to-end dual-stage, multi-level semantic information learning architecture. Specifically, a topic-level semantic feature extraction module is incorporated to discern the topic distribution of the text. Initially, this module synergizes the text's hidden state, acquired from the Bi-GRU module, with the topic extractor for joint learning of local-global semantic information in the text. MSF-Net employs a multi-attention module to proficiently capture word relationships and semantic details by modeling the complete text sequence, informed by learned topic and context information, thereby aiding the model in paraphrase identification. Comparative and ablation experiments on the extensive LCQMC text dataset are presented in this paper. The MSF-Net model achieves precision, recall, F1 score, and accuracy rates of 78.69, 94.14, 85.72, and 87.13, respectively. The results substantiate MSF-Net's superiority over baseline models in capturing semantic information and reinforcing paraphrase recognition tasks.**

***Index Terms*—paraphrase identification, topic inference, attention mechanism, semantic information**

I. Introduction

Paraphrase identification is an important task in the field of natural language processing, which aims to automatically recognize the paraphrase of a given sentence from the text. Textual paraphrase refers to two sentences that express the same semantic information, but are described using different words or sentence structures. Paraphrase identification is crucial for many applications, such as machine translation, question-answering systems, and automatic summarization. Identifying paraphrases in the text can help improve the accuracy and naturalness of these application systems. In recent years, with the development of deep learning technology, paraphrase identification has been widely applied, and more and more researchers have started to study this topic.

Paraphrase identification is a challenging task due to the intricacies of language and the diversity of writing styles. Paraphrases can take diverse forms, involving alterations in vocabulary, syntax, and semantics, which poses a significant challenge in creating an accurate and robust paraphrase identification system. Table I illustrates six examples where researchers attempted to discern the distinctions between the original sentences and their paraphrased counterparts, considering word meanings' particularity and the semantic information of the entire sentence.

In recent years, there has been a growing interest in paraphrase identification, leading to the emergence of various methods to tackle this issue. These approaches range from traditional rule-based techniques to more advanced machine learning methods, including neural networks and deep learning. Despite notable progress in the field, several challenges persist, such as handling idiomatic expressions, addressing subtle contextual differences, and recognizing paraphrases across different languages. Therefore, further research is imperative to develop more effective and efficient paraphrase identification techniques.

TABLE I
Examples of the LCQMC Dataset

Original Sentence	Paraphrase Sentence	Positive(1) /Negative(0)
淘宝账号冻结怎么办? If a Taobao account is suspended, what can be done to resolve the issue?	什么都没做淘宝账号就被冻结了? Taobao account has been suspended without any actions taken by the account holder.	0
哺乳期可以用哪些面膜? Which types of facial masks are safe to use during lactation?	膜夕瘦脸面膜哺乳期可以用吗? Is it safe to use the Mo Xi slimming facial mask during lactation?	0
如何能让胡子长的慢点? What are some ways to slow down the growth of facial hair?	怎么才能让自己脸上不长痘痘啊? What are some effective strategies for preventing acne on the face?	0
怎样写英文摘要? How to write an English abstract?	英文摘要怎么写? Strategies for Writing an Effective English Abstract	1
云数贸是什么？是传销嘛？ What is Yunshuimao? Is it a pyramid scheme?	云数贸是传销吗？ Is Yunshuimao a pyramid scheme?	1
这个女神是谁？ Who is this goddess?	女神是谁？ Who is the goddess?	1

The architecture of neural networks enables the modeling

Manuscript received May 8, 2023; revised September 4, 2023. The research work was supported by Liaoning Key Laboratory of Intelligent Construction Internet of Things Application Technology (2021JH13/10200051).

Wenrui Xue is a graduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 935843919@qq.com)

Yujun Zhang is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author, e-mail: 1997zyj@163.com)

Xinyu Wang is a graduate student of School of Computer Technology,Dalian Maritime University,Dalian,116000, China. (e-mail: 1710050555@qq.com)

Shuting Ge is a graduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: geshuting22@163.com)

of grammar and semantics in a sentence by integrating smaller, fundamental semantic units such as words and phrases to generate more complex semantic units like sentences. Figure 1 illustrates the fundamental process of sensation recognition, which is supported by deep learning. However, if a term is rarely used but holds significant meaning based on its context, the model may struggle to accurately interpret its meaning in certain situations. This limitation arises because deep learning models heavily rely on extensive training data, and words that occur infrequently in the training data are not typically well-learned.

Furthermore, traditional models suffer from poor interpretability, making it challenging to understand why certain results are obtained using these models. On the other hand, deep learning models tend to be opaque and fail to provide transparent explanations or justifications for their outcomes, which may restrict their applicability in certain domains.

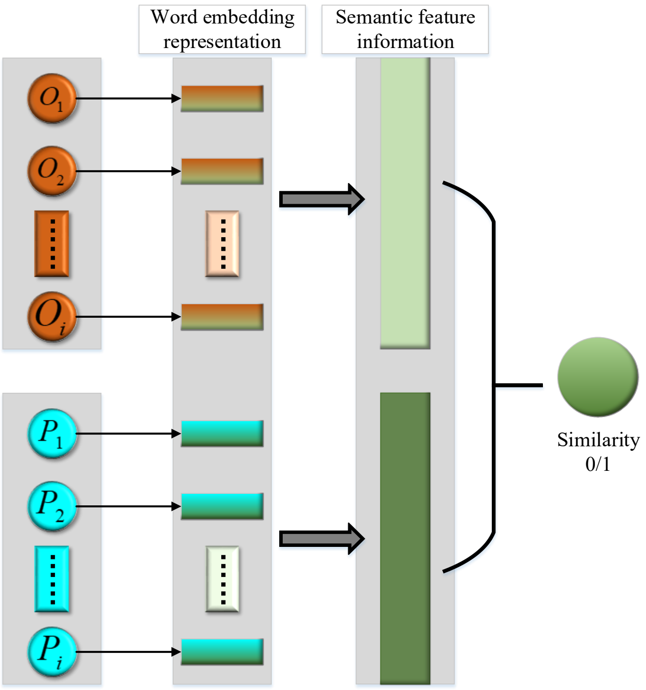

Fig. 1 Flowchart of the Paraphrase Identification

In order to resolve these concerns, this paper presents the following main contributions,

1) We propose a multi-level semantic feature network extractor for the paraphrase identification (MSF-Net) model.

2) Development of the topic-level semantic feature extraction module (TSFM) within MSF-Net. The TSFM employs a topic extractor to identify latent topics and assigns a topic probability distribution to each document, thus learning the global semantic information of the text. Additionally, it combines with a Bi-GRU module to learn hidden states in each sentence, capturing local-global semantic information.

3) Implementation of the multi-attention module (MAM) in MSF-Net, which utilizes a multi-head attention mechanism to weigh the entire sequence and extract global semantic information. This information is then fused with the topic-level semantic information and local semantic information learned by the TSFM module through an attention mechanism layer, resulting in a comprehensive and accurate semantic representation.

4) Conducting comparative and ablation experiments on the LCQMC dataset to evaluate the performance of the MSF-Net model in sentence-matching tasks. The results demonstrate that MSF-Net surpasses all baseline models, effectively learning global, local, and topic-level semantic information of the text and exhibiting promising practical applications.

II. Related Work

Understanding the underlying semantic information of two phrases and trying to differentiate between their respective meanings is the primary focus of the subtask of text semantic analysis known as sense recognition. This falls within the broader context of text semantic analysis. Initially, sensation identification approaches mainly focused on lexical matching or manually created corpora. WordNet[1] is an example of a lexical matching tool for sense identification that may infer word meaning by establishing associations between words, such as synonyms, hypernyms, and hyponyms. This kind of tool can be used to recognize senses. Context-based and semantic role labelling-based approaches have steadily been hotspots for study as machine learning and deep learning have become more popular. The methodologies based on machine learning, such as naive bayes and support vector machines, were brought into the field of sensation recognition. At the beginning of the 2010s, neural network-based methodologies such as convolutional neural networks (CNNs) [2] and recurrent neural networks (RNNs) [3] started to emerge as prominent areas of study interest. RNNs have shown excellent performance in modeling sentence similarity. Most RNNs focus on modeling based on the current sentence pair's hidden state, but the contextual information from another sentence in the hidden state generation process has not been well-researched. To address this, Chen[3] proposed a context alignment RNN (CA-RNN) model, which merges aligned word context information into a sentence pair for internal hidden state generation. To solve the definition recognition problem in the tax consulting field, Researchers have found that text datasets often lack training inputs for conventional classifiers, requiring significant effort to develop the quality of instance representations and language knowledge on which the model relies, and some text is overly identified for understanding. Mohamed[4] present a hybrid method for identifying sentence paraphrases, his method solves the challenge of determining the degree to which two sentences are semantically similar when both sentences include named things. The suggested method differentiates the calculation of semantic similarity between named-entity tokens and the rest of the sentence text.

Jain[5] introduced a neural network architecture known as a capsule network, which utilizes discrete clusters[6] of neurons called capsules. This type of neural network is designed to determine the presence or absence of a specific entity in an instance. Dinh's[7] approach for English-Vietnamese cross-language paraphrase identification using hybrid feature classes may have limitations in terms of scalability to larger datasets and other language pairs.

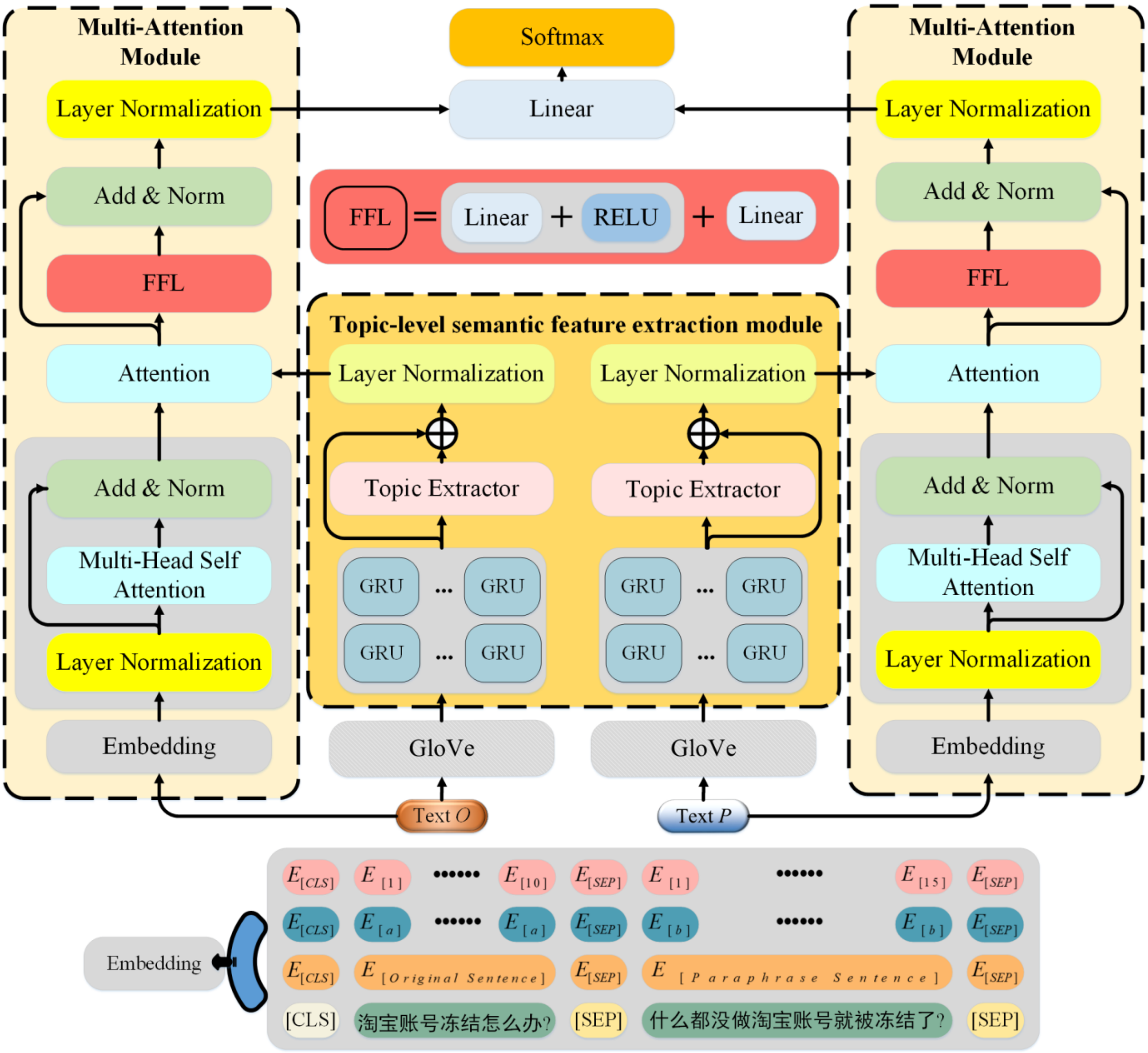
Multi-Attention Module
Layer Normalization
Add & Norm
FFL
Attention
Add & Norm
Multi-Head Self Attention
Layer Normalization
Embedding
Softmax
Linear
FFL = Linear + RELU + Linear
Topic-level semantic feature extraction module
Layer Normalization
Layer Normalization
Topic Extractor
Topic Extractor
GRU ... GRU
GRU ... GRU
GRU ... GRU
GRU ... GRU
GloVe
GloVe
Text O
Text P
Multi-Attention Module
Layer Normalization
Add & Norm
FFL
Attention
Add & Norm
Multi-Head Self Attention
Layer Normalization
Embedding
Embedding
E[CLS] E[1] •••••• E[10] E[SEP] E[1] •••••• E[15] E[SEP]
E[CLS] E[a] •••••• E[a] E[SEP] E[b] •••••• E[b] E[SEP]
E[CLS] E[Original Sentence] E[SEP] E[Paraphrase Sentence] E[SEP]
[CLS] 淘宝账号冻结怎么办? [SEP] 什么都没做淘宝账号就被冻结了? [SEP]

Fig.2 The Structure of MSF-Net Model. The MSF-Net model consists of a multi-head attention mechanism layer, a context encoder, a topic extractor, and an attention mechanism with topic semantic information. The specific workflow and equation calculation process of the MSF-Net model are given below.

Futhermore, the fuzzy-based method used to calculate feature classes may not always capture the nuances of the language and may lead to incorrect paraphrase identification. Ta's GAN-BERT[8] approach for paraphrase identification may have limitations in terms of the effectiveness of the noise filtering methods, which may lead to inaccurate paraphrase identification in certain cases. In addition, the effectiveness of the approach on other datasets beyond Mexican Spanish may need to be explored further. Xu's LSSE[9] learning model for paraphrase identification with lexical, syntactic, and sentential encodings may have limitations in terms of its computational complexity and potential difficulty in interpretability. Additionally, while the model improves upon the baseline, its improvement may not be sufficient for certain applications. Arase's[10] transfer fine-tuning of BERT[11] with phrasal paraphrases may have limitations in terms of its reliance on phrasal paraphrases, which may not always be available or reliable. Moreover, while the approach may improve BERT's performance on certain tasks, its performance on other tasks may need to be explored further. Finding out whether or not two sections of text have the same meaning is one of the steps involved in the process of discovering paraphrases. As a consequence of this, it is a highly significant component in a broad range of applications, including computer-assisted translation[12], question answering[13], machine translation[14, 15], and other similar applications.

III. MODEL

A. Problem Definition and Overview

Paraphrase identification can be formally defined as follows: given a data triple (O, P ,y), where O=(o_1, o_2,…, o_3), P=(p_1 ,p_2 ,…,p_3) and the label y typically takes values of $\{0,1\}$ where y=1 indicates that sentences O and P have the same meaning, while $y = 0$ indicates the opposite in the sense recognition task. Text matching[16] models are typically given a set of texts O and another set of texts P.

The model compares the degree of matching between each sentence o_i and its corresponding sentence p_i, ranks the degree of matching for each pair of sentences, and the higher the degree of matching value corresponds to the higher level of semantic equivalence between p_i and o_i. Finally, a softmax layer can be used to estimate the probability $P\left(y \mid O,P\right)$, for example, in the task of question-answering selection, the answer text with the highest degree of matching should be selected as the matching result.

B. Topic-level semantic feature extraction module

In the Topic-level semantic feature extraction module, we first input Text *O* and Text *P* into the GloVe[17] layer for static encoding. Each character is converted into a 300-dimensional word vector representation. Then, we use a Bi-GRU[18] module to encode each text by inputting the text sequence into a bidirectional GRU model, which obtains a hidden state sequence for each time step. As a gated recurrent unit model, GRU can calculate the current time step's hidden state based on the input and the previous time step's hidden state, thus capturing local semantic information in the sequence.

Specifically, we consider each text as a word sequence with lengths of n and m, respectively, where each word is represented by a *d*-dimensional word vector. We arrange these word vectors in sequence to obtain an $n \times d$ matrix O and an $m \times d$ matrix P. For each time step *t*, the input gate formula is defined as shown equation(1), the update gate formula is defined as an equation(2), the candidate hidden state formula is defined as an equation(3), the hidden state update formula is defined as an equation(4), and the output gate formula is defined as the equation(5).

$$r_t = \sigma(W_r[e_t, h_{t-1}] + b_r), \tag{1}$$

$$z_t = \sigma(W_z[e_t, h_{t-1}] + b_z), \tag{2}$$

$$\tilde{h}_t = tanh(W_h[e_t, r_t \odot h_{t-1}] + b_h), \tag{3}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \tag{4}$$

$$y_t = W_y h_t + b_y, \tag{5}$$

where $W_r, W_z, W_h, W_y, b_r, b_z, b_h, b_y$ represents the trainable model parameters. During the training process, the values of these parameters are adjusted iteratively to optimize the model's performance on a given task. ⊙ denotes element-wise multiplication, σ represents the sigmoid function, and $tanh$ represents the hyperbolic tangent function.

In summary, we take matrices *O* and *P* as input and feed them into a bidirectional GRU model, which produces two hidden state sequences: h_1, representing the hidden state sequence of the forward GRU, and h_2, representing the hidden state sequence of the backward GRU. The forward GRU is defined by the following formula, while the backward GRU is defined by the following formula,

$$h_{\{1,t\}} = GRU\left(o_t,\ h_{\{1,t-1\}}\right), \tag{6}$$

$$h_{\{1,t\}} = GRU\left(p_t,\ h_{\{1,t-1\}}\right), \tag{7}$$

$$h_{\{2,t\}} = GRU\left(o_t,\ h_{\{2,t-1\}}\right), \tag{8}$$

$$h_{\{2,t\}} = GRU\left(p_t,\ h_{\{2,t-1\}}\right), \tag{9}$$

where $h_{\{1,0\}}$ represents the initial hidden state of the forward GRU, and $h_{\{2,n+1\}}$ represents the initial hidden state of the backward GRU. In the formulas, o_t represents the *t*-th row of the input matrix *O*, which corresponds to the word vector of the *t*-th word. Similarly, p_t corresponds to the word vector of the *t*-th word in the input matrix *P*. Finally, we use h_1 and h_2 as the output of the Bi-GRU module, which are then used for the subsequent model training and inference.

For a set of Text *O*, *P* pairs where each document is composed of several word items, denoted as $d_i = \{w_{i1}, w_{i2}, ..., w_{I,m_i}\}$, where w_{ij} represents the *j*-th word item in the *i*-th document, the core idea of the topic extractor[19] model is to assign each word item in each document to different topics, where each topic is composed of a set of topic words. The topic extractor assumes that there are *K* topics for the text *O*, *P*, denoted as $K = \{\theta_1, \theta_2, ..., \theta_K\}$. The probability of assigning the *j*-th word item in the *i*-th document to the *k*-th topic is denoted as $p(z_{ij} = k \mid d_i)$, where z_{ij} represents the assigned topic of the *j*-th word item in the *i*-th document.

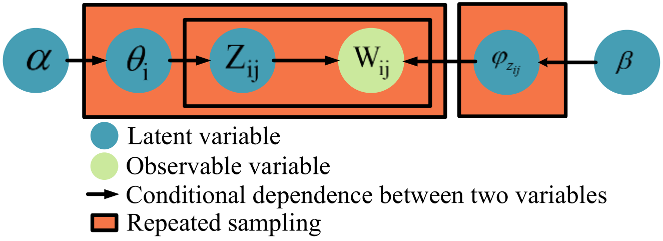

Fig. 3 The Topic Extractor Structure Diagram

For each word $w_{i,j}$ in a document d_i, its corresponding topic can be viewed as being randomly sampled from the topic distribution θ_i, and then the specific word $w_{i,j}$ is generated based on the topic's word distribution ϕ_k. Specifically, we can define a latent variable $z_{i,j}$ to represent the topic of $w_{i,j}$, and generate $w_{i,j}$ based on the distributions θ_i and ϕ_k. For each topic $k \in \mathbb{R}^K$ and word $w_{i,j}$, their generation probabilities can be represented as $p\left(w_{i,j} |\ w_{i,j} = k\right) = \phi_{k,w}$ and $p\left(w_{i,j} = k |\ d_i\right) = \theta_{i,k}$. Then, the generation process of $w_{i,j}$ in a document d_i can be represented as follows (10),

$$\begin{aligned} p\left(w_{i,j} |\ d_i\right) &= \sum_{k=1}^{K} p\left(w_{i,j} |\ w_{i,j} = k\right) p\left(w_{i,j} = k |\ d_i\right) \\ &= \sum_{k=1}^{K} \phi_{k,w_{i,j}} \theta_{i,k}, \end{aligned} \tag{10}$$

In the equation above, *K* represents the number of topics. The mathematical definition of the probability of topic k_i in document d_i is shown in the equation(11),

$$P(d_i = k_i) = \frac{n_{d_i,k_i} + \alpha}{\sum_{j=1}^{K} n_{d_i,j} + \alpha \times K}, \tag{11}$$

where n_{d_i,k_i} is the number of words assigned to topic k_i in document d_i. The hyperparameter a is drawn from a Dirichlet distribution, controlling the smoothness of topic allocation. The probability of word w_{ij} in topic k_i is mathematically defined as shown in the equation (12),

$$P(w_{i,j} = w \mid z_i = k_i) = \frac{n_{w_i,k_i} + \beta}{\sum_{v=1}^{V} n_{k_i,v} + \beta \times V}, \tag{12}$$

where the variable n_{w_i,k_i} represents the number of times the word $w_{i,j}$ appears in topic k_i . Similarly, β is a hyperparameter of the topic extractor that follows a Dirichlet prior distribution, which controls the degree of smoothing of word assignments. The variable *V* denotes the size of the vocabulary.

We introduce two prior distributions in the topic extractor, namely the topic distribution (document-topic distribution) θ and the vocabulary distribution (topic-word distribution) ϕ . The topic distribution θ represents the probability distribution of each topic in each document, while the vocabulary distribution ϕ represents the probability distribution of each vocabulary in each topic. These two distributions can be considered hyperparameters of the topic extractor. The topic extractor determines them through cross-validation. The mathematical formalization of the topic extractor is shown in equation (13):

$$p(\theta, \phi, z, w \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{k=1}^{K} p(\phi_k \mid \beta) \prod_{i=1}^{m} p(w_i \mid z_i, \phi_{\{1:K\}}), \tag{13}$$

In this equation, α and β are the parameters of the Dirichlet distributions, used to represent the prior distributions of the document-topic distribution θ and topic-word distribution ϕ , respectively. $p(\theta \mid \alpha)$ is a Dirichlet distribution used to generate the topic distribution θ_i of each document d_i . $p(\phi_k \mid \beta)$ is a Dirichlet distribution used to generate the word distribution ϕ_k of each topic *k*. $p(z_i \mid \theta)$ is a multinomial distribution used to generate the topic z_i of each word w_i . $p(w_i \mid z_i, \phi_{1:K})$ is a multinomial distribution used to generate the probability of each word w_i , with conditional probability ϕ_{z_i,w_i} . To train the Topic Extractor, we need to estimate the parameters $\alpha, \beta, \theta, \phi$, and z from the observed words in the text, to obtain the optimal topic-word distribution.

During training, we maximize the probability of the topic assignments for all documents in the collection. The logarithmic likelihood function is calculated using the following formula:

$$\mathcal{L}(\alpha, \beta) = \sum_{d=1}^{D} \sum_{k=1}^{K} n_{d_i,k_i} \log \frac{n_{d_i,k_i} + \alpha}{\sum_{j=1}^{K} n_{d_i,j} + \alpha \times K} + \sum_{k=1}^{K} \sum_{w=1}^{V} n_{w_i,k_i} \log \frac{n_{w_i,k_i} + \beta}{\sum_{v=1}^{V} n_{k_i,v} + \beta \times V}, \tag{14}$$

Finally, we combine the sentence-level representations h_1 and h_2 with the topic representations z_O and z_p to obtain the overall representations v_O and v_p for the input texts *O* and *P*, respectively.

$$v_O = Concat(h_O, z_O), \tag{15}$$

$$v_y = Concat(h_y, z_y), \tag{16}$$

where the term $Concat(\bullet)$ represents the concatenation operation, which joins the sentence-level representation and the topic representation along a specific dimension. By doing so, we can use v_O and v_y to calculate their similarity.

C. Multi-Attention Module

The main idea behind the multi-attention module is to use a multi-head attention mechanism network and a topic-aware attention layer to learn the local features of text *O* and text *P*.

The embedding layer converts each word in the input text sequence into a corresponding vector representation. The length of the input text sequence is *n*, and each word is represented by a *d*-dimensional one-hot vector. The weight matrix of the embedding layer is denoted as E, and its size is *V*×*d*, where *V* is the size of the vocabulary. Therefore, the calculation formula of the embedding layer is as follows,

$$Embedding(x_i) = TokenEmb(x_i) + SegmentEmb(x_i) + \text{Position}Emb(x_i), \tag{17}$$

where, x_i represents the one-hot vector of the *i*-th word in the input text sequence. $TokenEmb(\bullet)$, $SegmentEmb(\bullet)$, and $\text{Position}Emb(\bullet)$ represent token embedding, parity embedding, and location embedding.

The Layer Normalization layer performs normalization on each dimension of each word vector in each sample. The input word vectors are denoted as $h = [h_1, h_2, ..., h_n]$, where h_i represents the i -th word vector. The computation of the Layer Normalization layer is as follows,

$$LayerNorm(h_i) = \gamma_i \frac{h_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i, \tag{18}$$

where the γ_i and β_i terms are learnable parameters used to scale and shift the i -th word vector, respectively. The μ_i and σ_i terms represent the mean and standard deviation of the i -th word vector across the batch, while ϵ is a small constant added to avoid division by zero.

The multi-attention module utilizes the multi-head attention[20] layer to compute self-attention on the input text sequence and concatenates the resulting multi-head attention vectors. The input word vectors are mapped to three different spaces (query, key, and value) using three linear transformations (W_q , W_k , and W_v). The dot product between the query and key is calculated to obtain the attention score (attention weight matrix), which is then used to weight the values, resulting in a multi-head attention vector. Finally, the multi-head attention vectors are concatenated along the last dimension to obtain the final multi-head attention vector. The calculation formula for the Multi-Head Attention layer is as follows,

$$\text{MultiHead}(h) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h) W_o, \tag{19}$$

where the head_i represents the *i*-th head attention vector, and W_o is a weight matrix used to map the multi-head attention vectors to the final output space. For the *i*-th head attention vector, the calculation formula is as follows:

$$head_i = Attention\left(hW_q^{(i)}, hW_k^{(i)}, hW_v^{(i)}\right), \tag{20}$$

where the attention layer uses the weight matrices $W_q^{(i)}$, $W_v^{(i)}$, and $W_v^{(i)}$ to map the query, key, and value vectors, respectively. The *Attention* function computes the attention scores and performs weighted summation.

To calculate the attention vector, the Attention layer takes in the hidden state representations *vo* and *vp* with the thematic information and the text sequence that has been processed by Add&Norm. The Attention layer performs the following steps: 1) Linear transformation of the word vector sequence to obtain the query vector; 2) Linear transformation of *vo* and *vp* to obtain the key and value vectors, respectively; 3) Dot product computation between the query and key vectors to obtain the attention scores (attention weight vector), followed by weighted summation between the attention weight vector and the value vector to obtain the weighted sum vector. The calculation formula of the Attention layer is as follows,

$$Attention(h, vo, vp) = \sum_{i=1}^{n} \alpha_i v_i, \tag{21}$$

where h_i denotes the *i*-th word vector, v_o and v_p represent the hidden state representations with theme information, α_i represents the attention weight of the *i*-th word vector, and v_i denotes the value vector of the *i*-th word vector.

In the Multi-Attention Module, the Feed Forward layer performs a non-linear transformation on the attention vectors that are computed. The Add&Norm layer then adds the attention vectors obtained from the Attention layer to those obtained from the Feed Forward layer and applies Layer Normalization. The Linear layer then maps the normalized attention vectors to a new space. Finally, the Softmax[21] layer performs a softmax operation on the output of the Linear layer to obtain the final output result.

IV. EXPERIMENT

In this section, we conduct experiments on a real large-scale Chinese data set to verify the validity of the MSF-Net. The experimental data set, baseline model, parameter setting and evaluation index are introduced respectively.

A. Experimental Datasets and Evaluation Measures

To validate the effectiveness of the MSF-net model, this paper conducted comparative and ablation experiments on the LCQMC dataset[22]. The LCQMC dataset is a large-scale Chinese question-matching corpus proposed by Liu et al. in 2018. Compared to other datasets, the LCQMC dataset focuses more on intent matching rather than paraphrasing. Liu et al. collected a large number of question pairs related to high-frequency words from various fields using search engines, and then filtered out unrelated pairs in LCQMC using Wasserstein distance, resulting in 26,068 question pairs. The LCQMC dataset includes 238,766 question pairs in the training set, a development set with 8,802 question pairs, and a test set with 12,500 question pairs. The distribution of the LCQMC dataset is shown in Table II.

The job of identifying paraphrases is a binary classification issue that consists of detecting whether or not two parts of the text have the same meaning. Because the classification accuracy and efficacy of the model are of the highest relevance for this kind of work, accuracy is often employed as the key assessment parameter. The ratio of the number of samples that were properly categorized to the total number of samples is what constitutes accuracy. This ratio provides a direct measurement of the overall classification accuracy of the model.

TABLE II
Specific Statistics for the LCQMC Dataset

Dialogue dataset	train	dev	test
Number of data (pair)	238766	8802	12500
Positive sample (pair)	138574	4402	6250
Negative sample (pair)	100192	4400	6250

However, accuracy by itself may not be enough to adequately represent the performance of the model, particularly when there is an imbalance in the number of positive and negative samples. In the process of identifying paraphrases, there is often an imbalance between the positive and negative samples. This is due to the fact that the number of synonymous and nearly synonymous samples is much lower than the number of non-synonymous and non-nearly synonymous samples. In situations like this, the model may have an inherent bias toward predicting the bigger class of samples, but this would not be reflected in its accuracy. As a result, different measures including as accuracy, recall, and F1-score are often utilized in order to assess the effectiveness of the model when it comes to classification. Precision is determined by comparing the number of real true positives to the total number of predicted positives, while recall is determined by comparing the number of actual true positives to the total number of actual positives. The F1-score is an all-encompassing assessment measure that combines precision and recall, and it is able to take into account both the accuracy and the recall of the model at the same time.

$$ACC = (TP+TN)/(TP+TN+FP+FN), \tag{22}$$

$$Precision = TP/(TP+FP), \tag{23}$$

$$Recall = TP/(TP+FN), \tag{24}$$

$$F1-score = \frac{2\times(Precision\times Recall)}{(Precision+Recall)}, \tag{25}$$

where TP stands for true positive, which refers to the number of samples that were accurately predicted to be positive; TN stands for true negative, which refers to the number of samples that were accurately predicted to be negative; FP stands for false positive, which refers to the number of samples that were incorrectly predicted to be positive; and FN stands for false negative, which refers to the number of samples that were incorrectly predicted to be negative.

B. Baselines and Implementation Details

In this research, 11 different types of deep learning models are validated on the LCQMC data set. This is done so that the validity of MSF-NET may be examined. The choice of models is instructive of The Times overall aesthetic. The following list provides information that is specific to each model.

- CBOW[23]: The continuous bag-of-words (CBOW) is a word embedding technique based on the bag-of-words model that represents the semantic meaning of a current word as the average vector of its surrounding context words. Due to the lack of contextual information, this method encounters significant difficulties in semantic recognition tasks despite its impressive performance with large corpora. Specifically, CBOW cannot understand the semantic relationships between lengthy text sequences, resulting in severe problems with long-term dependencies.

- CNN[24, 25]: Each sentence is encoded as an embedding matrix that is input into a CNN. In the CNN model's convolutional layer, multiple convolutional kernels are used to learn local features. These characteristics possess translational invariance, allowing for enhanced recognition of sentences with semantically similar meanings. The convolutional layer of the CNN model performs sliding convolution with a fixed-size window. As a consequence, the model can only learn local features of fixed length, making it difficult to represent long-distance dependency relationships.
- BiLSTM[26]: The BiLSTM model is capable of capturing the sequential information of words within a sentence and learning long-range context information, making it more suitable for modeling the sentence's semantics. As a framework for end-to-end learning, the BiLSTM model can immediately acquire the optimal representation from the original data. BiLSTM dynamically adjusts to various sentence lengths, enabling it to manage inputs of variable lengths more effectively. Due to its approach of word-by-word processing, the BiLSTM model may encounter computational and memory issues when coping with extremely lengthy texts.
- BiMPM[27]: Wang present a bilateral multi-perspective matching (BiMPM) model for sentence matching in natural language. Our model encapsulates two sentences with a BiLSTM encoder and matches them from multiple perspectives in two orientations. Using another BiLSTM layer, the matching results are consolidated, and a layer that is completely interconnected makes the final determination.
- DFF[28]:This paper proposes a deep feature fusion model for sentence semantic matching (SSM), which incorporates an attention mechanism to capture the semantic context without sacrificing significant sentence encoding features. Model components include an embedding layer, a deep feature fusion layer, a matching layer, and a prediction layer. To preserve indistinguishable instances during the training process, a novel hybrid loss function is also proposed, which combines MSE and cross entropy.
- HiDR[29]:Yu present HiDR, a novel SSM model based on hierarchical 2D CNNs that can learn expressive sentence representation and capture inter-sentence interactions. The model employs bidirectional LSTMs to generate dimension-augmented representation and a sigmoidal function to output the degree of matching.
- FMSR[30]: Guo presented a frame-based multi-level semantics representation (FMSR) model to improve text matching neural computer systems. The FMSR model directly extracts multi-level semantic information from words using FrameNet frame and frame components. The FMSR model improves text matching by learning more accurate sentence representations using multi-level semantic information and attention processes.
- MGMSN[31]:Wang proposed a multi-granularity matching model based on Siamese neural networks to address the limitations of existing text matching algorithms. The model leverages both deep and shallow semantic similarity of input sentences to fully capture similar information between them. In addition, the model utilizes both word and character-level granularity in deep semantic similarity to handle the issue of out-of-vocabulary in sentences.
- Transformer[32]: Improving semantic representation, the Transformer model employs self-attention mechanisms to capture long-distance dependencies. Multi-head self-attention and residual connections contribute to the resolution of gradient vanishing and overfitting issues. The model is computationally intensive and may struggle to capture sequential information due to the use of unordered positional embeddings.
- BERT[33]: BERT is a pre-trained language model that encodes contextualized word embeddings for downstream natural language processing tasks using a transformer-based architecture. To employ BERT for sentence semantic matching, the input sentences are tokenized and then fed to a pre-trained BERT model to obtain contextualized embeddings. Then, these embeddings are input into a task-specific model, such as a completely connected layer or a BiLSTM, to perform sentence matching.
- ALBERT[34]: Lan propose two techniques for minimizing parameters to improve the scalability of BERT by reducing memory consumption and training time. In addition, we introduce a self-supervised loss to enhance inter-sentence coherence modeling and demonstrate that it benefits downstream tasks with multi-sentence inputs.

During our experiment, we utilized a vocabulary containing 50,000 terms that were frequently present in the dataset. For terms not included in the lexicon, we used the special token "UNK." The word vectors were assigned a dimension of 300, and the hidden vectors for Bi-GRU were set to a size of 200. A batch size of 32 was used in this experiment. To optimize the loss and update the parameters, we employed the Adam method with an initial learning rate of 0.0001. To avoid the parameters being overfitted during the training phase, we used the dropout strategy with a dropout rate of 0.3. This assisted us in reducing the problem of overfitting.

C. Comparing Experimental Results

From Table III, the MSF-Net model proposed in this paper has demonstrated improvements compared to the baseline models mentioned earlier on the LCQMC dataset.

Although CBOW can preserve the word order, it generates word vectors using more contextual information, which results in its inability to filter out some irrelevant issues. The CNN model cannot adequately capture long-distance dependencies and contextual information, and it is also prone to confusion when dealing with semantically similar but not identical words. The Bi-LSTM model classifies sequences by learning features from them, making it sensitive to noise in the sequence. All three models only focus on single-level semantic information, such as word-level semantics, fixed-length sentence semantics, or

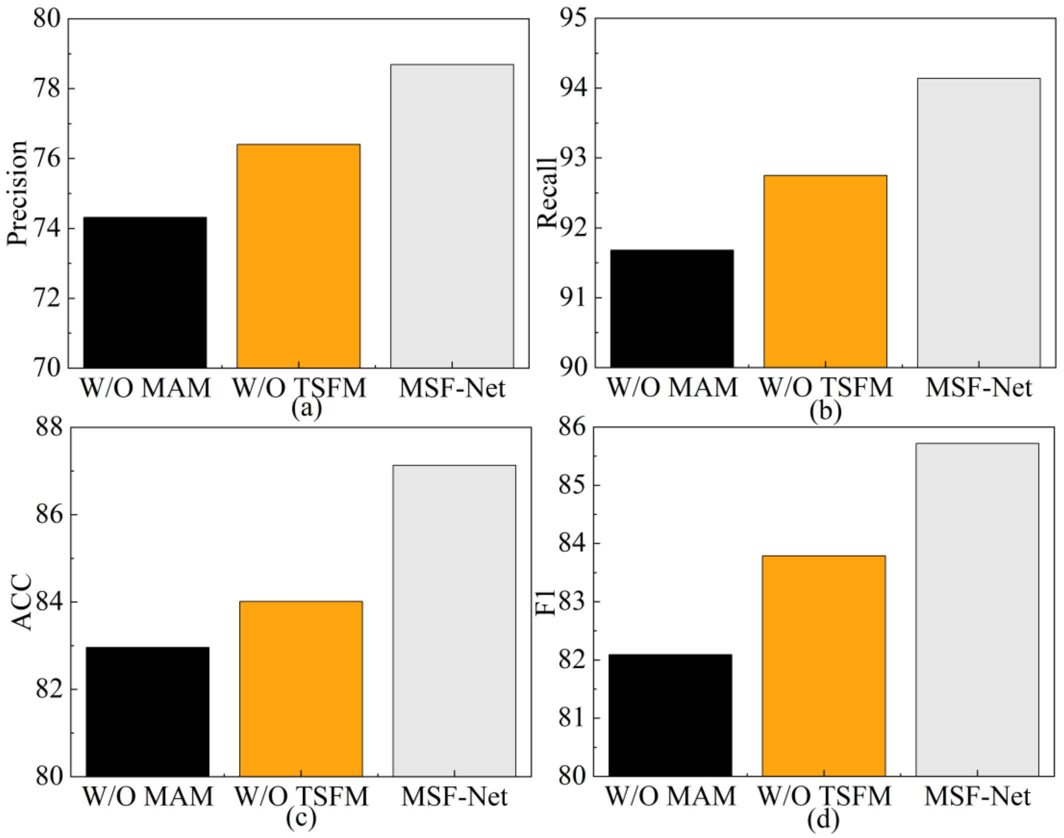

Fig. 4 The Ablation Experiments for MSF-Net Model

sentence-level semantics. Such a focus on semantic information is insufficient for the task of paraphrase identification, as evidenced by the experimental results in the first six rows of Table III.

The Bi-MPM and HiDR models utilize bidirectional matching networks, incorporating multiple matching strategies, including interactive matching and comparative matching, within a single model. However, this approach still falls under the classical recurrent neural network paradigm. Researchers have found that attention mechanisms can effectively learn both local and global semantic information from text in large-scale datasets. For paraphrase identification tasks, it is equally important to focus on both local and global information, as verified in baseline models such as DFF, FMSR, Transformer, BERT, and ALBERT. The multi-granularity semantic features learned by the MGMSN model are similar to those of the MSF-Net model proposed in this paper, which focuses on different levels of semantic feature information to obtain various feature vector representations. However, MSF-Net outperforms MGMSN in all four evaluation metrics, achieving 78.69% precision (P), 94.14% recall (R), 85.72% F1 score, and 87.13% accuracy (ACC). The performance of MSF-Net surpasses Transformer by 6.29/1.03/4.22/8.33 points, which is a remarkable result. This outstanding performance has also been validated in BERT and ALBERT models, which have increased the ACC metric by 0.23 and 0.37 points, respectively.

The MSF-Net model incorporates the concatenation of multiple feature maps and utilizes multiple self-attention mechanisms to fuse information, thereby enabling the extraction of multiple levels of features and enhancing the model's expressive capacity. Additionally, techniques such as label smoothing and class resampling are employed in MSF-Net to increase data diversity and enhance training robustness, leading to improved model performance.

TABLE III
The Performance of Different Models on LCQMC Dataset

Methods	P	R	F1	Acc
CBOW(c)	66.5	82.8	73.8	70.6
CBOW(w)	67.9	89.9	77.4	73.7
CNN(c)	67.1	85.6	75.2	71.8
CNN(w)	68.4	84.6	75.7	72.8
Bi-LSTM(c)	67.4	91.0	77.5	73.5
Bi-LSTM(w)	70.6	89.3	78.92	76.1
Bi-MPM(c)	77.6	93.9	85.0	83.4
Bi-MPM(w)	77.7	93.5	84.9	83.3
DFF(c)	78.58	93.88	85.51	84.15
DFF(w)	77.69	94.08	85.06	83.53
HiDR(c)	84.09	84.60	84.26	84.33
HiDR(w)	83.35	82.51	82.86	83.05
FMSR	-	-	-	79.0
MGMSN	-	-	-	85.0
Transformer	72.4	93.1	81.5	78.8
BERT	-	-	-	86.9
ALBERT	-	-	-	86.76
MSF-Net	**78.69**	**94.14**	**85.72**	**87.13**

D. Ablation Experimental Results

To validate the effectiveness of multi-level semantic features in semantic recognition tasks, we conducted ablation experiments on the MSF-Net model by evaluating its performance with and without specific components. We compared the full MSF-Net model with two variants: one without the topic-level semantic feature extraction module (W/O TSFM) and another without the multi-attention module (W/O MAM).

Figure 4 illustrates the results of the ablation experiments for the MSF-Net model. It is evident that both W/O TSFM and W/O MAM perform inferiorly compared to the full MSF-Net model across all four evaluation metrics. This clearly demonstrates that the multi-level semantic feature learning method effectively captures the local and global semantic features of the text, a finding supported by previous researchers.

Furthermore, the performance of W/O MAM is the lowest among the three variants, confirming the effectiveness of the attention mechanism, particularly on large-scale text data. By utilizing multiple layers of attention mechanisms, the model can effectively learn the semantic relevance between each pair of texts.

Although the performance of W/O TSFM is better than W/O MAM, it still lags behind the full MSF-Net model, highlighting the efficacy of the topic extractor in extracting latent topics from the text. The combination of the topic extractor and the Bi-GRU module learning method enables the capture of semantic feature relationships between local and global aspects of the text, thereby enhancing the model's performance on all four evaluation metrics.

V. Conclusion

In recent years, the widespread application of deep learning techniques in semantic recognition has been evident. However, challenges still remain due to inaccuracies in text semantics and difficulties in interpreting rare words, which limit the performance of deep learning models in processing such tasks. To tackle these issues, this paper introduces the Multi-level Semantic Feature Network Extractor (MSF-Net) model, which employs a two-stage multi-level semantic information learning approach that effectively captures word relationships and semantic information, resulting in improved semantic recognition performance. The model's evaluation on the LCQMC dataset shows promising results.

For future research, more precise methods for topic capture could be explored to enhance the model's understanding and expression capabilities in specific domains through probabilistic reasoning of topics. Additionally, this approach can be extended to other natural language processing (NLP) tasks to further boost the performance of deep learning models in the field of NLP.

References

[1] S. Cao, H. Vo, H. T.-T. Le, and D. Dinh, "Hybrid approach for text similarity detection in Vietnamese based on Sentence-BERT and WordNet," in *4th International Conference on Information Technology and Computer Communications, ITCC 2022, June 23, 2022 - June 25, 2022*, Virtual, Online, China, 2022: Association for Computing Machinery, pp. 59-63.

[2] J. Huang, D. Ji, S. Yao, and W. Huang, "Character-aware convolutional neural networks for paraphrase identification," in *23rd International Conference on Neural Information Processing, ICONIP 2016, October 16, 2016 - October 21, 2016*, Kyoto, Japan, 2016, vol. 9948 LNCS: Springer Verlag, pp. 177-184.

[3] Q. Chen, Q. Hu, J. X. Huang, and L. He, "CA-RNN: Using context-aligned recurrent neural networks for modeling sentence similarity," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018, February 2, 2018 - February 7, 2018*, New Orleans, LA, United states, 2018: AAAI press, pp. 265-273.

[4] M. Mohamed and M. Oussalah, "A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics," *LANGUAGE RESOURCES AND EVALUATION,* Article vol. 54, no. 2, pp. 457-485, 2020 JUN 2020.

[5] R. Jain, A. Kathuria, A. Singh, A. Saxena, and A. Khandelwal, "ParaCap: paraphrase detection model using capsule network," 2022, vol. 28: Springer Science and Business Media Deutschland GmbH, pp. 1877-1895.

[6] M. Ganardi and P. Gawrychowski, "Pattern Matching on Grammar-Compressed Strings in Linear Time," in *33rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, January 9, 2022 - January 12, 2022*, Alexander, VA, United states, 2022, vol. 2022-January: Association for Computing Machinery, pp. 2833-2846.

[7] D. Dien and T. Nguyen Le, "English-Vietnamese cross-language paraphrase identification using hybrid feature classes," *JOURNAL OF HEURISTICS,* Article vol. 28, no. 2, pp. 193-209, 2022 APR 2022.

[8] H. T. Ta, A. B. S. Rahman, L. Najjar, and A. Gelbukh, "GAN-BERT, an Adversarial Learning Architecture for Paraphrase Identification," in *2022 Iberian Languages Evaluation Forum, IberLEF 2022, September 20, 2022*, A Coruna, Spain, 2022, vol. 3202: CEUR-WS.

[9] S. Xu, X. Shen, F. Fukumoto, J. Li, Y. Suzuki, and H. Nishizaki, "Paraphrase Identification with Lexical, Syntactic and Sentential Encodings," *APPLIED SCIENCES-BASEL,* Article vol. 10, no. 12, 2020 JUN 2020, Art no. 4144.

[10] Y. Arase and J. Tsujii, "Transfer fine-tuning of BERT with phrasal paraphrases," *COMPUTER SPEECH AND LANGUAGE,* Article vol. 66, 2021 MAR 2021, Art no. 101164.

[11] X. Yu, Z. Li, J. Wu, and M. Liu, "Multi-module Fusion Relevance Attention Network for Multi-label Text Classification," *Engineering Letters,* vol. 30, no. 4, pp. 1237-1245, 2022.

[12] Y. Wang and W. Dong, "Application of Artificial Intelligence in Computer-Assisted English Vocabulary Translation," *Computer-Aided Design and Applications,* vol. 20, no. S5, pp. 32-41, 2023.

[13] S. Behmanesh, A. Talebpour, M. Shamsfard, and M. M. Jafari, "Improved relation span detection in question answering systems over extracted knowledge bases," *Expert Systems with Applications,* vol. 224, 2023.

[14] S. Tripathi and V. Kansal, "Error Classification and Evaluation of Machine Translation Evaluation Metrics for Hindi as a Target Language," in *19th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2018, March 18, 2018 - March 24, 2018*, Hanoi, Viet nam, 2023, vol. 13396 LNCS: Springer Science and Business Media Deutschland GmbH, pp. 331-345.

[15] S. K. Mondal, H. Zhang, H. M. D. Kabir, K. Ni, and H.-N. Dai, "Machine translation and its evaluation: a study," 2023.

[16] T. T. Huynh, T. Phamnguyen, and N. V. Do, "A Keyphrase Graph-Based Method for Document Similarity Measurement," *Engineering Letters,* vol. 30, no. 2, pp. 692-710, 2022.

[17] H. M. Balaha and M. M. Saafan, "Automatic exam correction framework (AECF) for the MCQS, essays, and equations matching," *IEEE Access,* vol. 9, pp. 32368-32389, 2021.

[18] G. Zhao, C. Zhang, H. Shang, Y. Wang, L. Zhu, and X. Qian, "Generative label fused network for imagetext matching," *Knowledge-Based Systems,* vol. 263, 2023.

[19] Z. Li, J. Wu, J. Miao, X. Yu, and S. Li, "A Topic Inference Chinese News Headline Generation Method Integrating Copy Mechanism," 2022.

[20] Z. Li, X. Xie, F. Ling, H. Ma, and Z. Shi, "Matching images and texts with multi-head attention network for cross-media hashing retrieval," *Engineering Applications of Artificial Intelligence,* vol. 106, 2021.

[21] Z. Li, J. Wu, J. Miao, X. Yu, and S. Li, "Multi-model Fusion Attention Network for News Text Classification," *International Journal for Engineering Modelling,* vol. 35, no. 2, pp. 1-15, 2022.

[22] X. Liu *et al.*, "LCQMC: A large-scale Chinese question matching corpus," in *27th International Conference on Computational Linguistics, COLING 2018, August 20, 2018 - August 26, 2018*, Santa Fe, NM, United states, 2018: Association for Computational Linguistics (ACL), pp. 1952-1962.

[23] W. Blacoe and M. Lapata, "A comparison of vector-based representations for semantic composition," in *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12, 2012 - July 14, 2012*, Jeju Island, Korea, Republic of, 2012: Association for Computational Linguistics (ACL), pp. 546-556.

[24] Y. Kim, "Convolutional neural networks for sentence classification," in *2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25, 2014 - October 29, 2014*,

Doha, Qatar, 2014: Association for Computational Linguistics (ACL), pp. 1746-1751.

[25] B. Agarwal, H. Ramampiaro, H. Langseth, and M. Ruocco, "A deep network model for paraphrase detection in short text messages," *Information Processing and Management,* vol. 54, no. 6, pp. 922-937, 2018.

[26] G. S. Tomar, T. Duque, O. Tackstrom, J. Uszkoreit, and D. Das, "Neural paraphrase identification of questions with noisy pretraining," in *EMNLP 2017 1st Workshop on Subword and Character Level Models in NLP, SCLeM 2017, September 7, 2017*, Copenhagen, Denmark, 2017: Association for Computational Linguistics (ACL), pp. 142-147.

[27] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *26th International Joint Conference on Artificial Intelligence, IJCAI 2017, August 19, 2017 - August 25, 2017*, Melbourne, VIC, Australia, 2017, vol. 0: International Joint Conferences on Artificial Intelligence, pp. 4144-4150.

[28] X. Zhang, W. Lu, F. Li, X. Peng, and R. Zhang, "Deep feature fusion model for sentence semantic matching," *Computers, Materials and Continua,* vol. 61, no. 2, pp. 601-616, 2019.

[29] R. Yu, W. Lu, Y. Li, J. Yu, G. Zhang, and X. Zhang, "Sentence Semantic Matching with Hierarchical CNN Based on Dimension-augmented Representation," in *2021 International Joint Conference on Neural Networks, IJCNN 2021, July 18, 2021 - July 22, 2021*, Virtual, Shenzhen, China, 2021, vol. 2021-July: Institute of Electrical and Electronics Engineers Inc., p. IEEE Computational Intelligence Society; International Neural Network Society.

[30] S. Guo, Y. Guan, R. Li, X. Li, and H. Tan, "Frame-based Multi-level Semantics Representation for text matching[Formula presented]," *Knowledge-Based Systems,* vol. 232, 2021.

[31] X. Wang and H. Yang, "MGMSN: Multi-Granularity Matching Model Based on Siamese Neural Network," *Frontiers in Bioengineering and Biotechnology,* vol. 10, 2022.

[32] A. Vaswani *et al.*, "Attention is all you need," in *31st Annual Conference on Neural Information Processing Systems, NIPS 2017, December 4, 2017 - December 9, 2017*, Long Beach, CA, United states, 2017, vol. 2017-December: Neural information processing systems foundation, pp. 5999-6009.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2019, June 2, 2019 - June 7, 2019*, Minneapolis, MN, United states, 2019, vol. 1: Association for Computational Linguistics (ACL), pp. 4171-4186.

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS," in *8th International Conference on Learning Representations, ICLR 2020, April 30, 2020*, Addis Ababa, Ethiopia, 2020: International Conference on Learning Representations, ICLR.