

Improving Text Semantic Similarity Modeling Through a 3D Siamese Network

Jianxiang Zang^{a,*} and Hui Liu^a

^aSchool of Statistics and Informatics, Shanghai University of International Business and Economics, China

ORCID ID: Jianxiang Zang <https://orcid.org/0000-0003-0236-8215>,

Hui Liu <https://orcid.org/0000-0003-1679-6560>

Abstract. Siamese networks have gained popularity as a method for modeling text semantic similarity. Traditional methods rely on pooling operation to compress the semantic representations from Transformer blocks in encoding, resulting in two-dimensional semantic vectors and the loss of hierarchical semantic information from Transformer blocks. Moreover, this limited structure of semantic vectors is akin to a flattened landscape, which restricts the methods that can be applied in downstream modeling, as they can only navigate this flat terrain to capture semantic nuances. To address this issue, we propose a novel 3D Siamese network for text semantic similarity modeling, which maps semantic information to a higher-dimensional space. The three-dimensional semantic tensors not only retains more precise spatial and feature domain information but also provides the necessary structural condition for comprehensive downstream modeling strategies to capture them. Leveraging this structural advantage, we introduce several modules to reinforce this 3D framework, focusing on three aspects: feature extraction, attention, and feature fusion. Our extensive experiments on four text semantic similarity benchmarks demonstrate the effectiveness and efficiency of our 3D Siamese Network.

1 Introduction

The aim of modeling text semantic similarity is to predict the degree of similarity between a pair of text sequences [15, 6, 24, 27, 2]. One of the popular methods for modeling text semantic similarity is the Siamese network, which employs dual Transformer encoders to encode two texts separately and fuse them together at the matching layer [18, 19]. The Transformer blocks used as encoders capture hierarchical semantic information that can be viewed from two aspects: positional (spatial) domain information and feature domain information [30]. Shallow Transformer blocks have high global feature encapsulation, whereas deep blocks have comparatively lower global feature encapsulation. On the other hand, deep Transformer blocks have high encapsulation of local feature and positional information, while shallow blocks have low encapsulation of local feature information.

In order to fully utilize these semantic information for modeling semantic similarity, researchers have introduced different modules into the models from three perspectives: feature extraction [26, 14, 9], late attention [16, 20, 2], and feature fusion [31, 6, 26, 18, 35, 19]. However, as illustrated in Figure 1(a), these methods are based on a

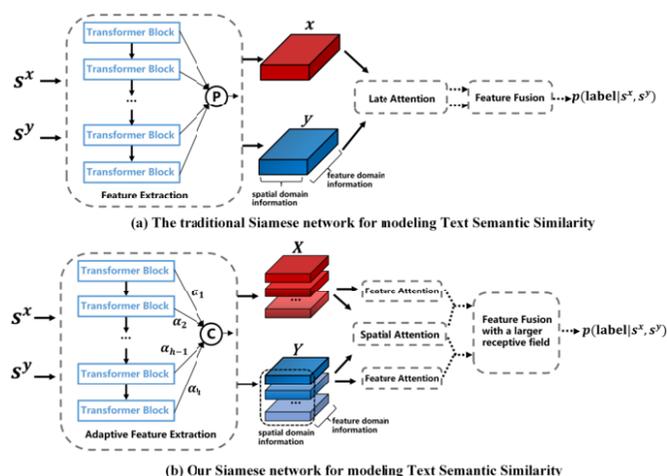


Figure 1: Comparison between the traditional Siamese Network and our 3D network for modeling text semantic similarity, where 'P' denotes the pooling operation and 'C' denotes the concatenation operation.

flattened processing, pooling the representations from each Transformer, and in some cases, even solely utilizing the representation from the final Transformer block. [18, 16, 22, 19]. These processes lead to two-dimensional (sentence length, feature dimension) semantic vectors and causes a substantial loss of hierarchical semantic information from different Transformer blocks. Moreover, this limited structure of semantic vectors is akin to a flattened landscape, which restricts the methods that can be applied in downstream modeling, as they can only navigate this flat terrain to capture semantic nuances. To tackle this issue, as depicted in Figure 1(b), we have implemented a novel technique that models text semantic similarity through a 3D Siamese network¹. This 3D network architecture maps semantic information to a higher-dimensional space, comprehensively and effectively retaining the hierarchical semantic information from Transformer blocks. For traditional methods of modeling two-dimensional semantic tensor similarity, the limitations lie in the tensor dimensions, which dictate the use of a single attention mechanism, and the reliance on global pooling during the feature fusion stage. In contrast, the resulted three-dimensional semantic tensors provides the necessary structural condition for a more diverse range of options for attention and feature fusion in down-

* Corresponding Author. Email: 21349110@suibe.edu.cn

¹ The code will be open-sourced: https://github.com/hggzjx/3D_SN

stream tasks, enabling a more comprehensive and robust capture of spatial and feature domain information. Leveraging this structural advantage, and drawing inspiration from modules within the image processing domain [13, 33, 12, 29, 21], we suggest several modules focusing on three vital aspects to reinforce this network: feature extraction, attention, and feature fusion. Specifically, we introduce Adaptive weighting to Extract Features (AFE) from the representation from each transformer block, and use a stacked concatenation method to encode sentences for downstream semantic similarity modeling. The 3D Siamese network provides increased potential for attention mechanisms and feature fusion. In contrast to traditional single late attention, we simultaneously incorporate Spatial Attention (SA) and Feature Attention (FA), forming a robust information interactor. Moreover, from a higher-dimensional standpoint, we draw upon Inception architecture and Dilated Convolution to develop a Receptive Field Module (RFM) for feature fusion.

Our main contributions resides in the following aspects:

- We propose a 3D Siamese network that comprehensively and effectively retaining the hierarchical semantic information from Transformer blocks, and provides the necessary structural condition for comprehensive downstream semantic similarity modeling strategies to capture them.
- To reinforce this 3D Siamese network, we suggest several modules for the downstream semantic similarity task that focus on three crucial aspects: feature extraction, attention, and feature fusion.
- Introduced modules exhibit "plug-and-play" characteristic, which contributes to the model's robust modularity and scalability.

2 Related Work

Text semantic similarity modeling is a core problem in Natural Language Processing, which aims to determine the semantic relationship between two given textual inputs[15, 6, 24, 27, 2].

In recent years, Transformer-based pre-trained encoders have garnered significant attention for modeling text semantic similarity [30], yielding remarkable results [25, 7, 34]. Two primary training paradigms have emerged, based on Transformer encoders. The first entails the joint encoding of sentence pairs by Transformer modules [23], which facilitates comprehensive interaction between sentence pairs in terms of information. However, this approach results in increased computational costs and higher inference latency, posing challenges for industrial deployment. Alternatively, the second approach employs a structure with Siamese Transformer encoders [7, 26, 18, 22]. This method enables offline computation of text embeddings, significantly reducing online latency [3, 26].

To enhance the performance of Siamese networks, researchers have proposed corresponding improvements in three areas: feature extraction, late attention, feature fusion. In terms of feature extraction, researchers [26] compared the effects of three pooling fusion methods on the performance of Siamese BERT. The average-pooling strategy can extract more abundant information compared with max pooling. In addition to pooling operation, DenseNet [14, 9] has also been used for feature extraction [3], largely preserving the information of the original text features. In order to enhance the model's interaction capability, researchers have introduced numerous late attention strategies, such as the cross-attention layer [16], the MLP layer [20], and the Transformer layer [2]. Regarding feature fusion, tensor cross-fusion [31, 6, 26] and pooling fusion [6, 26, 18] are the most commonly used methods for feature fusion. Based on these, [35] propose an improved mechanism for information fusion

and interaction that compares local and aligned representations from three perspectives. [19] propose VIRT-Adapted Interaction which performs feature extraction and information exchange simultaneously.

As mentioned in Section 1, these methods are based on a uniform and flattened processing, pooling the representation from each Transformer block during encoding, [26, 18, 16, 22, 19], the limitation of which lies in the insufficient extraction of these two types of information, as well as the constraints imposed on the downstream choices of similarity modeling. To address this issue, we have adopted a novel approach modeling text semantic similarity through a three-dimensional Siamese network and suggest several modules to reinforce it.

3 Methodology

The task of modeling text semantic similarity can be described as formulated in Equation 1. Given the input sentence pair s^x and s^y , the training goal of the model is to train a classifier ξ that computes conditional probabilities $P(\text{label}|s^x, s^y)$ to predict the relationship between the output sentence pairs based on the output probabilities. The label $\in \Omega$ represents different degrees of semantic similarity.

$$P(\text{label}|s^x, s^y) = \xi(s^x, s^y) \quad (1)$$

In this paper, we introduce a novel 3D Siamese network for text semantic similarity modeling, emphasizing feature extraction, attention, and feature fusion. The 3D Siamese network consists of the following modules we introduced: Adaptive Feature Extraction (AFE), Spatial Attention & Feature Attention (SA&FA), and Receptive Field Module (RFM). AFE comprehensively and efficiently preserves semantic information from different Transformer blocks. SA captures long-range dependencies between sentence pairs, while FA dynamically adjusts feature weights within certain sentence for better discrimination. We combine SA and FA for robust information interactor. Finally, RFM leverages the Inception architecture and Dilated Convolution layers to capture a larger receptive field.

3.1 Adaptive Feature Extraction

Compared to the traditional method of using pooling operations to compress semantic representations from transformer blocks,, we propose the use of trainable Adaptive weights for Feature Extraction (AFE) in each block. These weighted representations are concatenated in a three-dimensional form, delivering the most comprehensive information for downstream task modeling.

Given the input sentence pair $s^x = [s_1^x, s_2^x \dots s_{l_x}^x]$ and $s^y = [s_1^y, s_2^y \dots s_{l_y}^y]$, we use Transformer blocks to acquire representations of the input sentences. Assuming there are H Transformer blocks, the semantic representations of the sentence pair x_i^h and y_j^h can be obtained in the h^{th} Transformer block.

$$\begin{aligned} x_i^h &= \text{TransformerBlock}^h(s^x, i), i \in [1, 2 \dots l_x] \\ y_j^h &= \text{TransformerBlock}^h(s^y, j), j \in [1, 2 \dots l_y] \end{aligned} \quad (2)$$

Our goal is to obtain a semantic tensor comprising the representations from each Transformer block, concatenated using trainable weights. We define our unnormalized attention vectors \tilde{x}_i^h and \tilde{y}_j^h as formulated in Equation 3.

$$\begin{aligned} \tilde{x}_i^h &= \sigma(W_2^h(\text{ReLU}(W_1^h x_i^h + b_1^h)) + b_2^h) \\ \tilde{y}_j^h &= \sigma(W_2^h(\text{ReLU}(W_1^h y_j^h + b_1^h)) + b_2^h) \end{aligned} \quad (3)$$

Where $W_1^h, W_2^h, b_1^h, b_2^h$ are shared and trainable parameters for the representation from the h^{th} Transformer block. σ refers to the sigmoid activation function. ReLU refers to the ReLU activation function. We calculate the normalized attention weights α_i^h and β_j^h , as formulated in Equation 4.

$$\alpha_i^h = \frac{\tilde{x}_i^h}{\sum_{i=1}^{l_x} \tilde{x}_i^h}, \quad \beta_j^h = \frac{\tilde{y}_j^h}{\sum_{j=1}^{l_y} \tilde{y}_j^h} \quad (4)$$

These weights are applied to perform a weighted concatenation on the representations from each transformer block, producing final semantic tensors $\{\mathbf{X}, \mathbf{Y}\}$ that are utilized for further modeling of semantic similarity in downstream tasks.

$$\begin{aligned} \mathbf{X} &= [\alpha_1^1 \mathbf{x}_1^1; \alpha_1^2 \mathbf{x}_1^2; \dots; \alpha_i^H \mathbf{x}_i^H] \\ \mathbf{Y} &= [\beta_1^1 \mathbf{y}_1^1; \beta_1^2 \mathbf{y}_1^2; \dots; \beta_j^H \mathbf{y}_j^H] \end{aligned} \quad (5)$$

In above equation both \mathbf{X} and $\mathbf{Y} \in \mathbb{R}^{H \times L \times D}$. D is the dimension of the sentence vector. H is the number of Transformer blocks, and L is the length of the sentence vector. $[\cdot]$ is the concatenation operation.

3.2 Spatial Attention & Feature Attention

As we have mentioned, traditional methods of modeling text semantic similarity employ consistent pooling compression of the representations generated by transformer blocks. As a result, they can only adopt a single late attention to model the spatial and feature domain information between sentences. In contrast, our approach takes a 3D perspective and allows for the separate modeling of Spatial Attention (SA) and Feature Attention (FA). By fusing the semantic tensors that have been enhanced through these attention mechanisms for feature fusion, our approach forms a more powerful information interactor. Specifically, Spatial Attention effectively learns the inter-dependencies between various positions of two semantic tensors, enhancing its ability to capture long-range dependencies between sentence pairs. Meanwhile, Feature Attention is capable of discerning dependencies among features within the semantic tensor, dynamically adjusting the weights for each feature and thereby extracting more discriminative features. For the semantic tensors $\{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{H \times L \times D}$, we fuse the results of two attention-enhanced processes using either feature-level or element-wise multiplication to obtain the semantic tensors $\{\mathbf{X}', \mathbf{Y}'\} \in \mathbb{R}^{H \times L \times D'}$.

$$\begin{aligned} \mathbf{X}' &= \text{SA}(\mathbf{X}, \mathbf{Y}) \odot \text{FA}(\mathbf{X}) \\ \mathbf{Y}' &= \text{SA}(\mathbf{Y}, \mathbf{X}) \odot \text{FA}(\mathbf{Y}) \end{aligned} \quad (6)$$

where SA and FA refer to Spatial Attention Module and Feature Attention Module respectively. \odot refers to elements-wise multiplication.

3.2.1 Spatial Attention

Spatial Attention enables the modeling of inter-dependencies between different positions of semantic tensor pairs, which enhances the ability of our model to capture long-range dependencies between pairs of sentences. Let $\{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{H \times L \times D}$ denote the semantic tensors that have been processed by the Adaptive Feature Fusion and Extraction module. We first reshape them to \mathbf{K}_i and \mathbf{Q}_j with the shape of $N \times D$, where $N = H \times L$. Different from self-attention, our query matrix \mathbf{Q} and key matrix \mathbf{K} interact with each other for cross-sentence information communication. Therefore, as illustrated

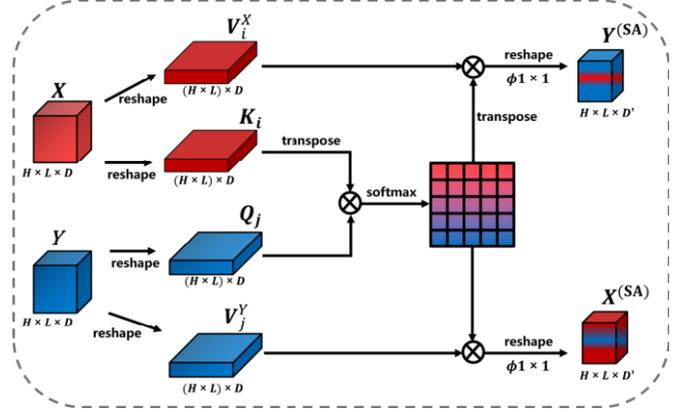


Figure 2: Overview of Spatial Attention.

in Figure 2, we design two branches to process sentence pair's representations respectively. After that, we perform a matrix multiplication between the transpose of \mathbf{K} and \mathbf{Q} and transpose this calculation result to obtain the semantic tensor of another branch. Finally, we apply a softmax layer on them to calculate M_{ji}^Y and M_{ij}^X respectively.

$$M_{ji}^Y = \frac{\exp(\mathbf{K}_i^T \cdot \mathbf{Q}_j)}{\sum_{k=1}^N \exp(\mathbf{K}_i^T \cdot \mathbf{Q}_k)}, \quad M_{ij}^X = \frac{\exp(\mathbf{K}_i^T \cdot \mathbf{Q}_j)}{\sum_{k=1}^N \exp(\mathbf{K}_k^T \cdot \mathbf{Q}_j)} \quad (7)$$

Here, similarity matrix M_{ji}^Y measures the \mathbf{Y} 's j^{th} position's impact on the i^{th} position of \mathbf{X} . The more similar the feature representations of the two positions are, the larger the correlation between them becomes. The function of M_{ij}^X is similar with Equation 7.

Meanwhile, we reshape \mathbf{X} and \mathbf{Y} reshape them to \mathbf{V}_i^X and \mathbf{V}_j^Y with the shape of $N \times D$, where $N = H \times L$. Then we perform a matrix multiplication between \mathbf{V}_j^Y and M_{ji}^Y , after reshaping them to the shape of $N \times D$. For another branch, similar to the above calculation process, we perform a matrix multiplication between \mathbf{V}_i^X and M_{ij}^X .

$$\begin{aligned} \mathbf{X}_i^{(\text{SA})} &= \sum_{j=1}^N M_{ji}^Y \cdot \mathbf{V}_j^Y, \quad i \in [1, \dots, N] \\ \mathbf{Y}_j^{(\text{SA})} &= \sum_{i=1}^N M_{ij}^X \cdot \mathbf{V}_i^X, \quad j \in [1, \dots, N] \end{aligned} \quad (8)$$

Where $\mathbf{X}_i^{(\text{SA})}$ is the weighted sum of $\{\mathbf{Y}_j\}_{j=1}^N$ with M_{ji}^Y . Intuitively, the purpose of Spatial Attention is to use the elements in $\{\mathbf{Y}_j\}_{j=1}^N$ that are related to $\mathbf{X}_i^{(\text{SA})}$ to represent $\mathbf{X}_i^{(\text{SA})}$. The same is performed for another branch with Equation 8. Finally, we apply reshape operations to the results and then leverage a 1×1 convolution layer to get the outputs $\mathbf{X}^{(\text{SA})}$ and $\mathbf{Y}^{(\text{SA})}$, where $\mathbf{X}^{(\text{SA})}, \mathbf{Y}^{(\text{SA})} \in \mathbb{R}^{H \times L \times D'}$ and D' is the feature dimension of low dimensional mapping space.

3.2.2 Feature Attention

The Feature Attention is a computational unit designed to learn the inter-dependencies between features. It takes a semantic tensor $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D] \in \mathbb{R}^{H \times L \times D}$ or $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_D] \in \mathbb{R}^{H \times L \times D}$ as input and produces transformed tensor $\mathbf{X}^{(\text{FA})}$ or $\mathbf{Y}^{(\text{FA})} \in \mathbb{R}^{H \times L \times D'}$ with augmented representations of the same size as $\mathbf{X}^{(\text{SA})}$ or $\mathbf{Y}^{(\text{SA})}$. Inspired by the Squeeze-and-Excitation Networks [13, 33, 12], Feature Attention can be modeled in two

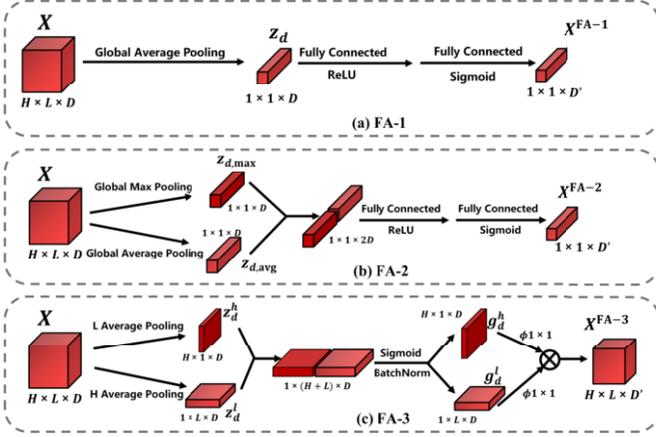


Figure 3: Overview of three forms of Feature Attention.

stages, the squeeze stage and the excitation stage. Specifically, the Squeeze stage is responsible for compressing the input semantic tensor to obtain a spatial context descriptor [11], which is a process of extracting global information. The Excitation performs non-linear transformations on the compressed semantic tensor to obtain a weight vector for each feature. We propose three forms of FA. To simplify illustration, we will only show the processing of FA in the branch of input X as example in Figure 3.

1. For FA-1, given the input X , to open up the correlation between feature and spatial domain information, we squeeze this semantic tensor using average pooling. The squeeze step for the d^{th} feature can be formulated in Equation 9.

$$z_d = \frac{1}{H \times L} \sum_{i=1}^H \sum_{j=1}^L \mathbf{x}_d(i, j) \quad (9)$$

Where z_d is the spatial context descriptor associated with the d^{th} feature. This squeeze operation makes collecting global information possible.

Next, we utilize the feedforward layers to obtain the semantic tensor $X^{(FA-1)} \in \mathbb{R}^{H \times L \times D'}$, fully capturing feature-wise dependencies.

$$X^{(FA-1)} = \sigma(W_2(\text{ReLU}(W_1 z + b_1)) + b_2) \quad (10)$$

where W_1, W_2, b_1, b_2 are trainable parameters, σ is the sigmoid function.

2. For FA-2, compared to FA-1, we argue that utilizing max pooling captures another crucial aspect of object characteristics, which enables us to infer Feature Attention with even greater precision. Firstly, we utilize average pooling and max pooling operations to generate two different spatial context descriptors: $z_{d,avg}$ and $z_{d,max}$.

$$z_{d,avg} = \frac{1}{H \times L} \sum_{i=1}^H \sum_{j=1}^L \mathbf{x}_d(i, j) \quad (11)$$

$$z_{d,max} = \max_{i=1}^H \max_{j=1}^L \mathbf{x}_d(i, j) \quad (12)$$

Similar to FA-1, Both descriptors are concatenated then forwarded to the feedforward layers to produce the final semantic tensor $X^{(FA-2)} \in \mathbb{R}^{H \times L \times D'}$.

$$X^{(FA-2)} = \sigma(W_2(\text{ReLU}(W_1([z_{d,avg}; z_{d,max}]) + b_1)) + b_2) \quad (13)$$

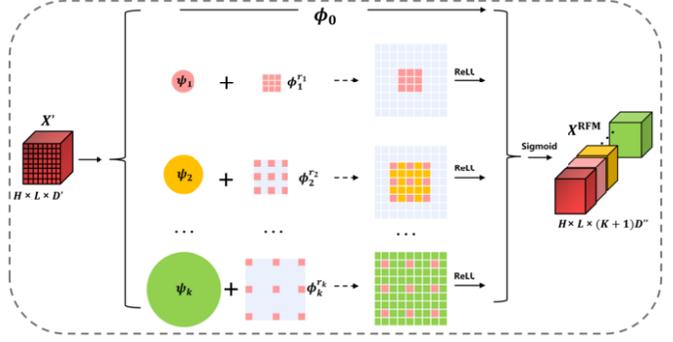


Figure 4: Overview of Receptive Field Module.

3. For FA-3, we further enhance the process by accurately model the relationships between features located at longer distances in a spatial context while also taking into account their specific positions within that space. To achieve this, we factorize the global pooling operation formulated in the original method into a pair of 1D feature encoding operations. Specifically, given an input X , we use two pooling kernels with different spatial extents, $(H, 1)$ and $(1, L)$, to encode each feature along the horizontal and vertical coordinates, respectively. This allows us to compute the output of the d^{th} feature at h^{th} Transformer block and the output of the l^{th} feature at the l^{th} position in the sentence, formulated in Equation 14.

$$z_d^h(h) = \frac{1}{L} \sum_{j=0}^L \mathbf{x}_d(i, j), \quad z_d^l(l) = \frac{1}{H} \sum_{j=0}^H \mathbf{x}_d(i, j) \quad (14)$$

The above two transformations enable a global receptive field and encode precise positional information. Compared to FA-1 and FA-2, we not only want to make full use of the captured spatial information but also to effectively capture inter-feature relationships. Specifically, given the descriptors produced by Equation 14, we apply sigmoid activation and batch normalization processing to the concatenated result.

$$\mathbf{g} = \text{BatchNorm}(\sigma([z_d^h; z_d^l])) \quad (15)$$

where $\mathbf{g} \in \mathbb{R}^{1 \times (H+L) \times D}$ is the intermediate descriptor that encodes spatial information in both the horizontal direction and the vertical direction. Here, D' is the feature dimension of low dimensional mapping. We then split \mathbf{g} along the spatial dimension into two separate tensors $\mathbf{g}^h \in \mathbb{R}^{H \times 1 \times D}$ and $\mathbf{g}^l \in \mathbb{R}^{1 \times L \times D}$. Another two 1×1 convolution layers ϕ_h, ϕ_l are utilized to reshape \mathbf{g}^h and \mathbf{g}^l to tensors with the same feature dimension to D' . Finally we leverage the matrix multiplication between these two tensor, and obtain $X^{(FA-3)} \in \mathbb{R}^{H \times L \times D'}$.

$$X^{(FA-3)} = \phi_h(\mathbf{g}^h) \cdot \phi_l(\mathbf{g}^l) \quad (16)$$

Where \cdot denotes the matrix multiplication.

3.3 Feature Fusion with a Larger Receptive Field

From our 3D perspective, we leverage a Receptive Field Module (RFM) to perform feature fusion. Inspired by the receptive field of the human vision [21]², we adopt the Inception architecture [29, 28, 21], which utilizes a multi-branch structure consisting

² The style of the illustrations follows the approach in reference [21].

Methods	Modules			Accuracy					Latency&Params	
	Feature Extraction	Attention	Feature Fusion	QQP	MRPC	SNLI	MNLI	Avg	Latency	Params
SBERT [26]	Pooling	-	Max&Avg Pooling	80.81	70.23	82.88	72.35	76.57	0.2ms	6.1M
ColBERT [18]	Pooling	LA	Max&Avg Pooling	86.31	74.23	86.12	76.23	81.22	0.3ms	6.1M
Our Networks	AFE	SA	Max&Avg Pooling	89.34	77.03	88.43	77.13	83.23	0.5ms	6.1M
	AFE	SA,FA-1	Max&Avg Pooling	89.45	77.04	89.10	77.11	84.06	0.5ms	6.1M
	AFE	SA,FA-2	Max&Avg Pooling	89.48	77.02	89.14	77.13	83.44	0.5ms	6.1M
	AFE	SA,FA-3	Max&Avg Pooling	89.46	77.04	89.02	77.21	83.43	0.5ms	6.1M
	AFE	SA	RFM	90.02	78.10	89.13	77.24	83.62	0.6ms	6.4M
	AFE	SA,FA-1	RFM	90.09	78.11	89.17	77.24	83.65	0.6ms	6.4M
	AFE	SA,FA-2	RFM	90.11	78.05	89.11	77.23	83.63	0.6ms	6.4M
	AFE	SA,FA-3	RFM	90.23	78.18	89.41	77.28	83.78	0.6ms	6.4M
BERT(interaction-based) [30]				91.51	87.81	90.01	86.50	88.96	27.4ms	107M

Table 1: Performance comparison on four benchmarks.

of convolutional layers with different kernel sizes. In addition, we use Dilated Convolutional layers [5, 4, 10], which have previously been employed in the segmentation algorithm Deeplab, to further expand the receptive field.

Figure 4 illustrates the multi-branch structure of RFM, where only the modeling of a single semantic tensor \mathbf{X}' is presented in the figure. Specifically, RFM takes the input semantic tensors $\mathbf{X}', \mathbf{Y}' \in \mathbb{R}^{H \times L \times D'}$ and constructs receptive field blocks by using multiple convolution kernels ψ of different sizes. This enables the network to capture feature information at different scales. Next, we perform Dilated Convolutions ϕ^r on $\psi(\mathbf{X}')$ and $\psi(\mathbf{Y}')$, which output multiple semantic tensors with different dilation rates. Assuming there are k different dilation rates for the convolution with the same shape, k corresponding semantic tensors can be obtained. In addition to Dilated Convolutions, a 1×1 convolutional layer can also be used to encode the feature vectors, resulting in a semantic tensor with global contextual information. This feature vector can help the model understand the global semantic information. Finally, all output feature vectors are concatenated to obtain the final results \mathbf{X}^{RFM} and $\mathbf{Y}^{\text{RFM}} \in \mathbb{R}^{H \times L \times (k+1)D''}$, which can be formulated in Equation 17 and Equation 18.

$$\begin{aligned} \mathbf{X}_i^{\text{RFM}} &= \text{ReLU}(\phi_i^{r_i}(\psi_i(\mathbf{X}'))), i \in [1, 2, \dots, k] \\ \mathbf{Y}_j^{\text{RFM}} &= \text{ReLU}(\phi_j^{r_j}(\psi_j(\mathbf{Y}'))), j \in [1, 2, \dots, k] \end{aligned} \quad (17)$$

$$\begin{aligned} \mathbf{X}^{\text{RFM}} &= \sigma([\mathbf{X}_1^{\text{RFM}}; \dots; \mathbf{X}_k^{\text{RFM}}; \phi_0(\mathbf{X}')] \\ \mathbf{Y}^{\text{RFM}} &= \sigma([\mathbf{Y}_1^{\text{RFM}}; \dots; \mathbf{Y}_k^{\text{RFM}}; \phi_0(\mathbf{Y}')] \end{aligned} \quad (18)$$

where ψ denotes the i^{th} convolution with different kernel size. $\phi_i^{r_i}$ denotes the i^{th} convolution with r_i dilation rates, ϕ_0 is the 1×1 convolution layer. The kernels among $\phi_i^{r_i}, i \in [1, 2, \dots, k]$ have the same size while the kernels among $\psi_i, i \in [1, 2, \dots, k]$ have different sizes to capture complex features.

Finally, we concatenate the two semantic tensors along with their element-wise multiplication results, and input them into a global average pooling layer and a fully connected layer to compute semantic similarity, where $\text{GAP}(\cdot)$ denotes the global average pooling operation.

$$\begin{aligned} \mathbf{v} &= \text{GAP}([\mathbf{X}^{\text{RFM}}; \mathbf{Y}^{\text{RFM}}; \mathbf{X}^{\text{RFM}} \odot \mathbf{Y}^{\text{RFM}}]) \\ P(\text{label} | \mathbf{s}^x, \mathbf{s}^y) &= \text{softmax}(W_2(\text{ReLU}(W_1(\mathbf{v}) + b_1)) + b_2) \end{aligned} \quad (19)$$

4 Experimental Setup

4.1 Datasets

We adopt the following four benchmarks for evaluation.

QQP (Quora Question Pairs) [17]: QQP is a dataset of around 400,000 question pairs from Quora. The goal is to determine if the questions have the same intent, assessing performance on semantic similarity and natural language understanding.

MRPC (Microsoft Research Paraphrase Corpus) [8]: MRPC has 5,800 English sentence pairs from online news. The task is to identify if the pairs are paraphrases, testing textual entailment and semantic similarity.

SNLI (Stanford Natural Language Inference) [1]: SNLI is a large textual entailment dataset with 570,000 English sentence pairs and labels for entailment relationships (entailment, contradiction, neutral). It evaluates natural language inference, determining if a hypothesis holds based on a premise.

MNLI (Multi-Genre Natural Language Inference) [32]: MNLI extends SNLI with 430,000 sentence pairs from various domains and genres. Like SNLI, it assesses natural language inference but covers a broader range of tasks. We evaluated on the mismatched development set of MNLI.

4.2 Baselines

We utilize three representative semantic similarity models as baselines.

SBERT [26]: Siamese BERT is a Siamese architecture that uses pre-trained BERT to separately produce embeddings of two inputs. The output embeddings of two sequences are concatenated to give final predictions.

ColBERT [18]: ColBERT serves as an efficient model for semantic similarity tasks, utilizing its late interaction mechanism to process queries and documents independently, enabling scalability. By employing dense vector representations, it effectively captures richer semantic relationships between text pairs.

BERT(interaction-based) [30]: As mentioned in Section 2, BERT(interaction-based) is an interaction-based model rather than utilizing a Siamese framework. Its underlying principle involves concatenating sentence pairs and encoding them through BERT for subsequent similarity prediction. While this approach enables rich interactions, it also introduces a significant number of training parameters and inference latency.

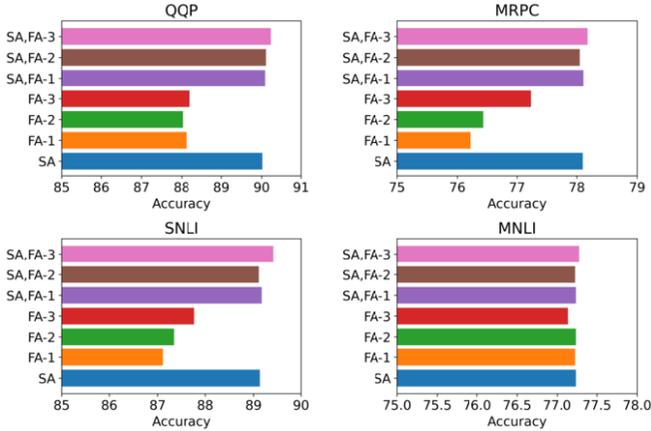


Figure 5: Results of ablation study on various combination of strategy selections for SA and FA.

4.3 Implementation Details

Our prediction target is $label^* = \text{argmax}P(\text{label}|s^x, s^y)$. For the QQP and MRPC datasets, $label^*$ represents {match, not match}, while for the SNLI and MNLI datasets, $label^*$ corresponds to {entailment, neutral, contradiction}. For all baselines, as well as our models, we use BERT (Base) as the encoder with 12 Transformer blocks, resulting in an encoded feature dimension of 768. Each sentence is padded to the average length of sentences in the dataset. To minimize the impact of the number of parameters on model performance, we adjusted the hyper-parameters of the convolutional and fully connected layers in the model, keeping the training parameters of all Siamese-based baselines and our model at similar levels. In the calculation of inference latency, the values we display are normalized by the test set size. We train the model for 15 epochs with a batch size of 32 and a learning rate of 0.001. Cross-entropy is used as the loss function, and the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is employed. We adopt a learning rate decay strategy and early stopping here. Specifically, if there is no early stopping after 10 epochs, the learning rate will be reduced with a decay rate of 0.1.

5 Results and Analysis

5.1 Main Results

Table 1 presents the main results for both baselines and our proposed approach across four evaluation tasks (QQP, MRPC, SNLI, MNLI). The principal difference between SBERT and CoBERT lies in the latter’s implementation of Late Attention (LA), which contributes to CoBERT’s enhanced performance. Our model is built upon various combination strategies of the aforementioned modules, utilizing proposed modules such as AFE for feature extraction, SA, FA-1, FA-2, and FA-3. In terms of feature fusion, our model explores both max pooling and average pooling, as well as RFM. The experimental results indicate that our model surpasses the baseline methods in terms of accuracy for the majority of tasks. Models employing RFM for feature fusion exhibit higher accuracy compared to those utilizing max pooling and average pooling. This can be attributed to the increased receptive field provided by RFM. Among the various attention mechanism combinations, the pairing of SA and FA-3 demonstrates the most favorable performance. BERT (interaction-based), as a powerful interactive-based model, outperforms our model in accuracy. However, its intricate interaction pattern leads to a substantially greater inference latency and parameter count relative to our method.

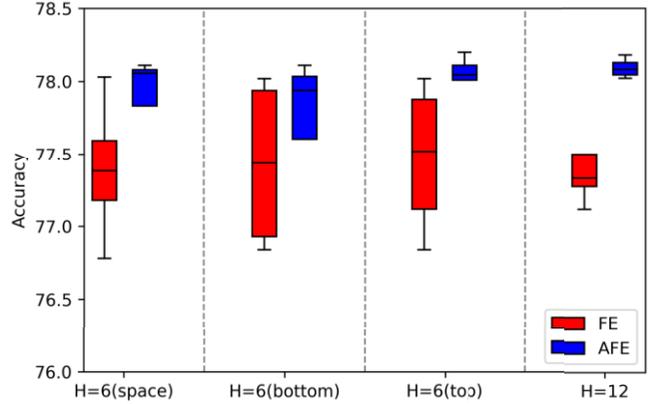


Figure 6: Robust experimental results on the impact of Transformer block and adaptive weight selection for feature extraction. FE and AFE represent feature extraction without the introduction of adaptive weights and feature extraction with the incorporation of adaptive weights, respectively.

In conclusion, our network demonstrates superior performance in terms of accuracy across four benchmark tasks when compared to the SBERT and CoBERT. Compared with BERT (interaction-based), our network exhibits significant advantages in aspects such as inference latency and parameter count. Consequently, our network requires less storage space and computational resources, rendering it more valuable for real-time applications and low-latency scenarios. The modular design of our network allows for the implementation of various combination strategies based on the selection of different components. These components possess minimal requirements for inter-module communication, facilitating seamless integration and enhancing the model’s modularity and scalability. This adaptable design paves the way for further optimization and refinement of the model, ultimately contributing to advancements in the field.

5.2 Ablation Study

5.2.1 Effect of SA&FA

Spatial Attention and Feature Attention together form a powerful information interactor. To gain a clear understanding of the impact of different attention combinations on model performance, we conducted ablation study on various combination strategy selections for SA and FA. As illustrated in Figure 5, the standalone SA consistently outperforms the independent FA-1, FA-2, and FA-3 across all tasks. For QQP, MRPC, and SNLI tasks, combining SA with FA-1, FA-2, and FA-3 results in a significant performance boost. In the absence of SA, the differences in performance among FA-1, FA-2, and FA-3 are not substantial. Upon incorporating SA, the SA and FA-3 combination achieves the best results in all four tasks. In summary, SA serves as the foundation for improving the modeling of semantic similarity, and the SA, FA-3 approach demonstrates more stable and superior performance compared to other methods. This suggests that combining SA and FA-3 methodologies may lead to better extraction and utilization of feature information in these tasks.

5.2.2 Effect of RFM

The superior performance of the RFM is due to the introduction of Inception architecture and Dilated Convolutions to increase the network’s receptive field. In this ablation study on QQP and MRPC

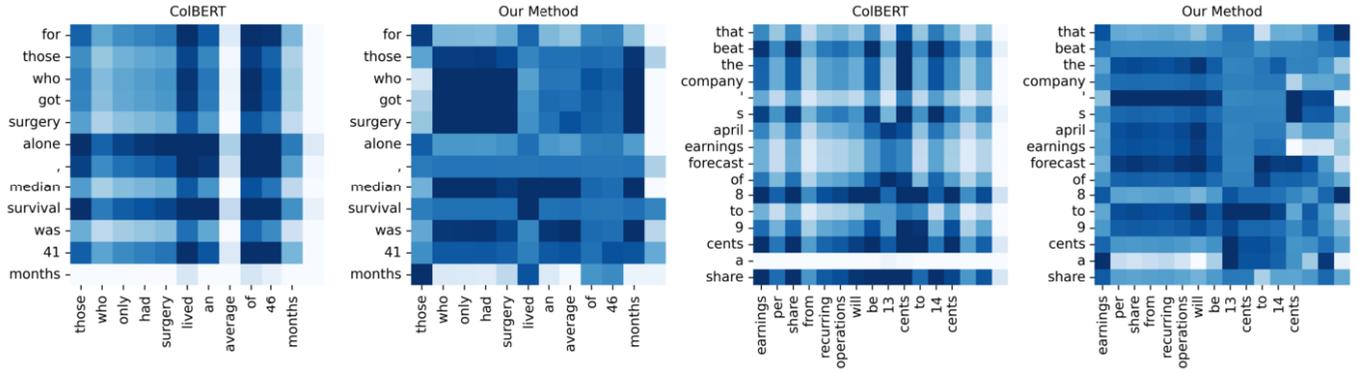


Figure 7: Both ColBERT and our method encode the similarity matrix of sentence pairs. The darker the color in the image, the higher the degree of similarity.

datasets, we investigate the impact of Inception architecture and Dilated Convolutions on the network’s capabilities. Specifically, for the Inception architecture, we only introduce multi-branch convolutions with different kernel sizes; for Dilated Convolutions, we only employ multi-branch Dilated Convolutions with different dilation rates. In the experiments, we use global max pooling and average pooling structures as references. As can be seen from Table 2, both Inception and Dilated Convolution are able to enhance the network’s performance.

Architecture	QQP	MRPC	Params
Max&Avg Pooling	89.46	77.04	6.1M
Inception	89.73(↑0.27)	77.86(↑0.82)	6.1M
Dilated Convolution	90.12(↑0.66)	77.91(↑0.87)	8.2M
RFM	90.23(↑0.77)	78.18(↑1.14)	6.4M

Table 2: Ablation study on different architectures in RFM.

5.3 Robustness Experiments

Our method demonstrates exceptional performance, which can be attributed to the efficient and comprehensive extraction of information from Transformer blocks. In this experiment, we investigated the impact of varying the number of retained Transformer blocks during feature extraction and the incorporation of adaptive weights on model performance. As reported in Figure 6, we conducted eight sets of experiments on MRPC dataset, examining the differences in experimental results after applying adaptive weights to four different model strategies with various Transformer block configurations: 6 spaced Transformer blocks (selecting Transformer blocks at intervals), 6 bottom Transformer blocks, 6 top Transformer blocks, and 12 Transformer blocks. When comparing different choices of Transformer blocks without introducing adaptive weights, we observed that selecting all Transformer blocks or retaining the top 6 Transformer blocks resulted in better model performance and stability than retaining the bottom 6 Transformer blocks or spacing 6 Transformer blocks. This is likely because the top Transformer blocks capture more comprehensive spatial and feature domain information after passing through multiple attention and feedforward layers. Introducing adaptive weights not only significantly enhanced the effectiveness of feature extraction but also minimized fluctuations between different network combination strategies. Furthermore, it reduced the influence of the number of Transformer blocks on the model, ultimately bolstering the network’s robustness.

5.4 Case Study

To illustrate the advantages of our 3D semantic similarity modeling framework compared to traditional models, we encoded sentences from the MRPC dataset using both the pre-trained ColBERT model and our own model, obtaining semantic similarity matrices for sentence pairs. Two examples are displayed in the Figure 7, along with visualizations of their attention similarity matrices. ColBERT can solely focus on parts of the sentence pairs with similar meanings, such as ‘survival’, ‘live’, ‘8 to 9 cents’, and ‘13 to 14 cents’. In contrast, our approach can pay more attention to key semantic information like ‘who got surgery’, ‘who only had surgery’, ‘average’, ‘median’, ‘beat the company’s April earnings cast’, and ‘earnings per share from recurring operations’. This demonstrates that our method, which efficiently utilizes raw information and models semantic similarity in a three-dimensional manner, can capture focus information more effectively.

6 Conclusion

In this paper, we propose a novel three-dimensional Siamese network for modeling semantic similarity. To reinforce this 3D framework, we have introduced a series of modules that address three key aspects: feature extraction, attention, and feature fusion. Extensive experiments on four text semantic similarity benchmarks demonstrate the effectiveness and efficiency of this 3D Siamese Network. Moreover, our introduced modules exhibit a “plug-and-play” characteristic, contributing to the model’s robust modularity and scalability.

In the future, we plan to apply this concept of three-dimensional semantic modeling to other tasks within the field of natural language processing.

References

- [1] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning, ‘A large annotated corpus for learning natural language inference’, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, (2015).
- [2] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian, ‘Deformer: Decomposing pre-trained transformers for faster question answering’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4497, (2020).
- [3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al., ‘Universal sentence encoder for english’, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 169–174, (2018).

- [4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, 'Deepplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs', *IEEE transactions on pattern analysis and machine intelligence*, **40**(4), 834–848, (2017).
- [5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, 'Encoder-decoder with atrous separable convolution for semantic image segmentation', in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, (2018).
- [6] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen, 'Enhanced lstm for natural language inference', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668, (2017).
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, '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, Volume 1 (Long and Short Papers)*, pp. 4171–4186, (2019).
- [8] Bill Dolan and Chris Brockett, 'Automatically constructing a corpus of sentential paraphrases', in *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, (January 2005).
- [9] Yichen Gong, Heng Luo, and Jian Zhang, 'Natural language inference over interaction space', in *International Conference on Learning Representations*.
- [10] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang, 'Context-aware feature generation for zero-shot semantic segmentation', in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1921–1929, (2020).
- [11] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid, 'Description of interest regions with local binary patterns', *Pattern recognition*, **42**(3), 425–436, (2009).
- [12] Qibin Hou, Daquan Zhou, and Jiashi Feng, 'Coordinate attention for efficient mobile network design', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13713–13722, (2021).
- [13] Jie Hu, Li Shen, and Gang Sun, 'Squeeze-and-excitation networks', in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, (2018).
- [14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, (2017).
- [15] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck, 'Learning deep structured semantic models for web search using clickthrough data', in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 2333–2338, (2013).
- [16] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston, 'Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring', in *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, (2020).
- [17] Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al., 'First quora dataset release: Question pairs. data. quora. com', (2017).
- [18] Omar Khattab and Matei Zaharia, 'Colbert: Efficient and effective passage search via contextualized late interaction over bert', in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, (2020).
- [19] Dan Li, Yang Yang, Hongyin Tang, Jingang Wang, Tong Xu, Wei Wu, and Enhong Chen, 'VIRT: improving representation-based models for text matching through virtual interaction', *CoRR*, **abs/2112.04195**, (2021).
- [20] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le, 'Pay attention to mlps', *Advances in Neural Information Processing Systems*, **34**, 9204–9215, (2021).
- [21] Songtao Liu, Di Huang, et al., 'Receptive field block net for accurate and fast object detection', in *Proceedings of the European conference on computer vision (ECCV)*, pp. 385–400, (2018).
- [22] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang, 'Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models', in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1864–1874, (2022).
- [23] Rodrigo Frassetto Nogueira and Kyunghyun Cho, 'Passage re-ranking with BERT', *CoRR*, **abs/1901.04085**, (2019).
- [24] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward, 'Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **24**(4), 694–707, (2016).
- [25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, 'Exploring the limits of transfer learning with a unified text-to-text transformer', *Journal of Machine Learning Research*, **21**, 1–67, (2020).
- [26] Nils Reimers and Iryna Gurevych, 'Sentence-bert: Sentence embeddings using siamese bert-networks', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, (2019).
- [27] Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych, 'Multicqa: Zero-shot transfer of self-supervised text matching models on a massive scale', in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2471–2486, (2020).
- [28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, 'Going deeper with convolutions', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, (2015).
- [29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, 'Rethinking the inception architecture for computer vision', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, (2016).
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', *Advances in Neural Information Processing Systems*, **30**, (2017).
- [31] Zhiguo Wang, Wael Hamza, and Radu Florian, 'Bilateral multi-perspective matching for natural language sentences', in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4144–4150, (2017).
- [32] Adina Williams, Nikita Nangia, and Samuel R Bowman, 'A broad-coverage challenge corpus for sentence understanding through inference', in *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pp. 1112–1122. Association for Computational Linguistics (ACL), (2018).
- [33] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, 'Cbam: Convolutional block attention module', in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, (2018).
- [34] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk, 'Approximate nearest neighbor negative contrastive learning for dense text retrieval', in *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, (2020).
- [35] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen, 'Simple and effective text matching with richer alignment features', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4699–4709, (2019).