Feature Differentiation and Fusion for Semantic Text Matching

Rui Peng, Yu Hong(✉), Zhiling Jin, Jianmin Yao, and Guodong Zhou

School of Computer Science and Technology, Soochow University, China
{rpeng124, tianxianer, zhljinjackson}@gmail.com, {jyao, guodongzhou}@suda.edu.cn

Abstract. Semantic Text Matching (STM for short) stands for the task of automatically determining the semantic similarity for a pair of texts. It has been widely applied in a variety of downstream tasks, e.g., information retrieval and question answering. The most recent works of STM leverage Pre-trained Language Models (abbr., PLMs) due to their remarkable capacity for representation learning. Accordingly, significant improvements have been achieved. However, our findings show that PLMs fail to capture task-specific features that signal hardly-perceptible changes in semantics. To overcome the issue, we propose a two-channel Feature Differentiation and Fusion network (FDF). It utilizes a PLM-based encoder to extract features separately from the unabridged texts and those abridged by deduplication. On this basis, gated feature fusion and interaction are conducted across the channels to expand text representations with attentive and distinguishable features. Experiments on the benchmarks QQP, MRPC and BQ show that FDF obtains substantial improvements compared to the baselines and outperforms the state-of-the-art STM models.

Keywords: Semantic text matching · Deep neural networks · Natural language processing

1 Introduction

STM is a fundamental and well-studied task of Natural Language Processing (NLP). It is defined as the task of determining the semantic consistency between texts. It has been applied for a wide range of downstream tasks. For example, in Community Question Answering (CQA), the STM model can be employed to retrieve the historical questions that are semantically equivalent to the queries.

Recently, the transformer-based PLMs have been leveraged to STM [24], playing the role of encoding sentences with attention mechanism (e.g., BERT [4], RoBERTa [14]). In general, they possess multi-layer transformers and learn to perceive and represent semantics from large corpora via well-designed self-supervised tasks. Transforming and fine-tuning PLMs have been proven effective in enhancing the current neural STM models.

The apparent contributions of PLMs for STM can be attributed to their profound perception of linguistic phenomena, as well as their awareness of a broader

Table 1: Examples of sentence pairs from the QQP corpus. The shared words of two sentences are marked in bold. A and B are the original sentences, while A' and B' denote the masked ones by replacing the shared words with a special token [MASK]. "Label" denotes the ground-truth label for matching, including $Paraphrase$ (i.e., *matched*) and $Non-paraphrase$. "*Predict*" indicates the prediction of a PLM-based STM model (BERT is used here), where the percentage numbers represent the prediction confidence levels.

A	What is the best course for learning **data structures**?
B	How can I learn **data structures** effectively?
A'	What is the best course for learning [MASK][MASK]?
B'	How can I learn [MASK][MASK] effectively?
Label	Non-paraphrases
Predict	Paraphrases (79.5%) → Non-paraphrases (100%)
A	**What are the requirements to** get into a **German** university?
B	**What are the requirements to** apply for **German** universities?
A'	[MASK][MASK][MASK][MASK][MASK] get into a [MASK] university?
B'	[MASK][MASK][MASK][MASK][MASK] apply for [MASK] universities?
Label	Paraphrases
Predict	Non-paraphrases (79.1%) → Non-paraphrases (54.6%)

range of commonsense knowledge. However, our findings show that PLMs fail to address the most challenging issue—anti-distraction. Distraction is caused by the repetitive contents occurring in a pair of sentences, which may easily distract an STM model. For example, the two instances in Table 1 separately provide a pair of sentences A and B, and each pair contains a large block of duplicated contents, such as "*learn data structures*" and "*what are the requirements to*". Such contents cause a close similarity between sentences, and therefore they are extra-distracting for determining semantic consistency.

In order to alleviate the distraction problem, we propose a two-channel Feature Differentiation and Fusion network (FDF). It is designed to highlight the features of non-repetitive contents in a sentence pair, with less information loss of attentive features in the repetitive contents.

Specifically, we conduct deduplication for the sentence pair by masking the shared tokens in them (see A' and B' in Table 1). This produces a pair of seemingly abridged sentences that merely possess non-repetitive contents. We utilize PLMs to extract token-level context-aware features for the original (i.e., unabridged) and abridged sentence pairs, through two separate channels. The resultant features are referred to as "shareable features" and "exclusive features" respectively. On this basis, Graph Convolutional Network (GCN for short) [11] is used to model interactions between shareable and exclusive features. Conditioned on the interaction strength, we additionally apply a gated layer to capture the important information of exclusive features, fusing that into shareable features. The goal is to produce distinguishable features with less information loss of attentive shareable features.

In our experiments, we combine the aforementioned shareable and exclusive features, and follow the conventional approaches to perform self-attention computation over them as well as pooling. Using the encoded features as reliance, we conduct binary classification (paraphrase or non-paraphrase) by a fully-connected linear layer with Softmax. Experiments are carried out over different benchmark corpora, including English MRPC [5] and QQP [9], as well as Chinese BQ [1]. Experimental results show that our method (FDF) yields substantial improvements compared to the PLM baselines BERT [4] and RoBERTa [14], where the most significant improvement is up to 2.1% accuracy rate (*Acc.*). Besides, FDF outperforms the state-of-the-art STM models.

The main contributions of this paper are concluded as follows:

- We propose to highlight the exclusive features under the condition that attentive information of shareable features is perceived and preserved.
- We construct a new PLM-based STM model (FDF) whose distinct components lay in the part of feature differentiation and fusion. Experiments show that FDF outperforms the existing STM models, over English and Chinese benchmark corpora.

2 Related Work

The previous work can mainly be divided into two categories: representation-based [10] and interaction-based approaches [4]. Representation-based models are generally constructed with the Siamese architecture, which encodes the considered sentences into embeddings separately, and decodes their relationship like semantic consistency by similarity estimation over embeddings. By contrast, interaction-based models straightforwardly involve interaction characteristics of sentences into the feature representation during encoding, instead of in the decoding phase. Our FDF can be sorted into the family of interaction-based models.

In order to perceive and represent deep features of texts for semantic matching, a variety of neural networks have been utilized at the earlier time, including CNN [8, 12], RNN [2, 17] and attention mechanism [10, 22]. Recently, PLMs have been further leveraged for SMT due to their remarkable success in boosting performance and versatility.

Specifically, Zhang et al. [23] incorporate a relation of relation classification task into their method to fully exploit the pairwise relation information. Their proposed method obtains the performance of 84.3% *Acc* on the MRPC corpus. Zou et al. [24] construct STM models merely using PLMs. Though, they develop a sophisticated and effective training strategy (namely divide-and-conquer training), where different losses are considered for optimizing the STM models, including KL-divergence loss, binary classification loss and distant supervision loss. Such training strategy contributes to feature differentiation and well-directed matching of concrete and abstract contents.

In addition, external knowledge has been used to enhance the PLM-based STM models. Liu et al. [13] train a semantic labeler over the external dataset

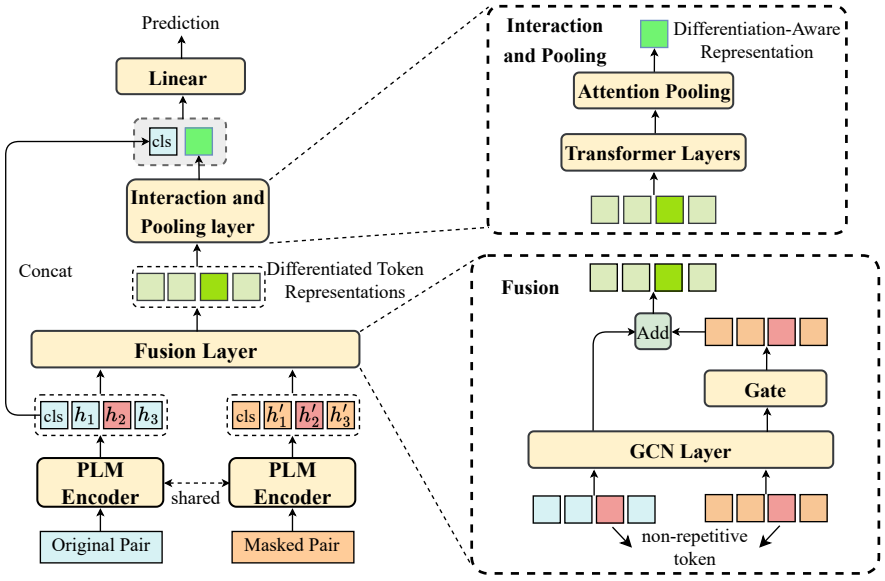


Fig. 1: Overview of the FDF network.

CONLL-2005, which provides semantic role embeddings for tokens. On this basis, they integrate semantic role embeddings with context-aware embeddings output by BERT. Xia et al. [21] incorporate synonym knowledge into BERT, enhancing word similarity perception at the self-attention computation stage.

3 Approach

First, let us give a formal definition of the STM task. Given two sentences $S_a = \{t_{a_1}, t_{a_2}, \cdots, t_{a_m}\}$ and $S_b = \{t_{b_1}, t_{b_2}, \cdots, t_{b_n}\}$ as the input, an STM model is required to output the binary decision about whether S_a and S_b are semantically consistent. Thus, STM can be boiled down to a binary classification task, grounded on the understanding of sentence semantics.

Our work concentrates on the encoding of sentences, providing reliable semantic features and representations for linear classification. The architecture of our model (FDF) is shown in Fig.1.

3.1 Input Layer

We concatenate the sentences S_a and S_b to form the original input sequence S_{ori}, i.e., $S_{ori} = \{[CLS], t_{a_1}, \cdots, t_{a_m}, [SEP], t_{b_1}, \cdots, t_{b_n}, [SEP]\}$, where $[CLS]$ and $[SEP]$ are specified as the special tokens. We regard S_{ori} as the unabridged sentence pair. Duplication detection is conducted to recognize the shared tokens in S_a and S_b (i.e., mutually-repetitive contents). Further, we uniformly replace

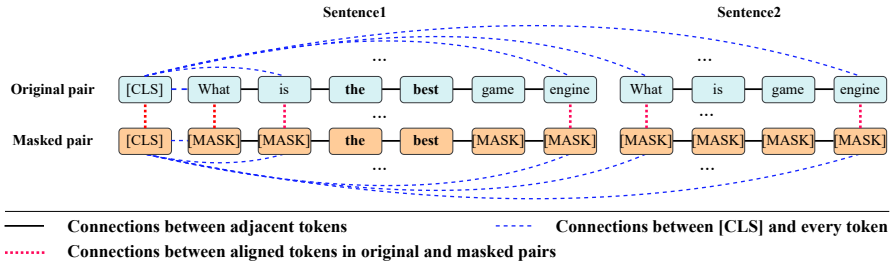


Fig. 2: An example of our proposed graph.

the shared tokens with the [MASK] tokens. This results in the production of the masked abridged sentence pair, which is denoted with S_{mask}:

$$S_{mask} = \{[CLS], t'_{a_1}, \cdots, t'_{a_m}, [SEP], t'_{b_1}, \cdots, t'_{b_n}, [SEP]\}$$
$$t'_i = \begin{cases} [MASK], & if \ \ t_i \in S_a \wedge \ t_i \in S_b \\ t_i, & Otherwise \end{cases} \tag{1}$$

Subsequently, we employ a PLM-based encoder to separately extract token-level features from the unabridged S_{ori} and abridged S_{mask}: $H = \text{PLM}(S)$. In this way, we obtain the shareable features H_{ori} and exclusive features H_{mask}.

3.2 Fusion Layer

We suggest that some exclusive features may signal the most distinguishable difference between the sentences S_a and S_b. Though, they cannot be used solely but cooperatively with the shareable features. Therefore, we fuse the two kinds of features (i.e., H_{ori} and H_{mask}), so as to avoid information loss throughout the semantics representation process, meanwhile preserving the positive effects of exclusive features.

We employ GCN [11] to fuse the features, in terms of the local interactive relationships of tokens in a predefined graph. Specifically, we construct a graph $G = (\mathcal{V}, \mathcal{E})$ using each token t_i ($t_i \in S_a \cup S_b$) as a node v_i ($v_i \in \mathcal{V}$). The edges $\mathcal{E}$ connecting the nodes are defined as follows: (1) Every node is connected to itself; (2) If two tokens in the unabridged sentence pair S_{ori} or the abridged sentence pair S_{mask} are adjacent, we connect them with an edge; (3) The special token $[CLS]$ of S_{ori} is connected with all the tokens in S_{ori} itself, while the special token $[CLS]$ of S_{mask} is connected with all the tokens (including [MASK]s) in S_{mask} itself; (4) Each token in S_{ori} is connected with the corresponding token (including [MASK]) in S_{mask}. We provide an example of $G = (\mathcal{V}, \mathcal{E})$ in Fig.2, where the lines indicate the connections between nodes.

We build the graph in this manner primarily for the following reasons. Employing multi-layer transformer architecture, PLMs are able to capture the critical words in sentences. However, they are deficient in perceiving local information

[23], which can also be instrumental to the matching process. To better model the local information, we adopt the second condition to gather the adjacent representations of each token. Besides, we design the third condition to enhance each token representation by global representation $[CLS]$ and vice versa.

The last condition enables the GCN module to model interactions between H_{ori} and H_{mask}. On the one hand, without the impact of shared tokens, the representations of non-repetitive tokens in H_{mask} only carry their semantic information. As a result, the message propagation between the nodes of non-repetitive tokens in H_{ori} and H_{mask} can enhance the representations of non-repetitive contents. On the other hand, during the encoding phase of PLM, the [MASK] tokens can only gather the contextual information from the tokens which are not masked (i.e., non-repetitive tokens). Therefore, their representations can weaken the corresponding representations of the repetitive tokens in H_{ori}, which can also be seen as an enhancement of non-repetitive token representations.

After constructing Graph G, we introduce its adjacency matrix $A \in \mathbb{R}^{2N \times 2N}$, where N is the sequence length of both S_{ori} and S_{mask}. Then we apply GCN to get the updated node features $\tilde{H}_{ori}$ and $\tilde{H}_{mask}$ as follows:

$$\tilde{H}_{ori}, \tilde{H}_{mask} = ReLU(\tilde{A}[H_{ori}; H_{mask}]W) \tag{2}$$

Here, $[\cdot;\cdot]$ denotes the concatenation operation, $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric matrix. D is the degree matrix of A, $D_{ii} = \sum_j A_{ij}$. $ReLU(\cdot)$ is the activation function, and $W \in \mathbb{R}^{d \times d}$ is a trainable weight matrix.

After the GCN module, each node is fully updated by aggregating the representations of its neighbors. Then we design a gated module to dynamically integrate the token representations $\tilde{h}_i^{ori}$ and their counterparts $\tilde{h}_i^{mask}$. Specifically, we first compare the two representations and calculate the score g to decide how to combine them, which can typically be conducted as follows:

$$\tilde{h}_i = [\tilde{h}_i^{ori}; \tilde{h}_i^{mask}] \tag{3}$$

$$g_i = tanh(w_g \tilde{h}_i + b_g) \tag{4}$$

where $tanh(\cdot)$ is the tanh activation function, $w_g \in \mathbb{R}^{d \times 1}$ and b_g are trainable parameters. Note that the g in Eq.(4) is a scalar score. Thus if we use it to integrate the two representations, all the dimensions are treated equally. Since the representation space is anisotropic [6] and each dimension represents different information, it is more reasonable to assign different scores to each dimension to achieve better fusion. Inspired by Shen et al. [18], we propose a multi-dimensional gated module. Instead of calculating a single scalar score, we calculate a feature-wise score matrix G_i, which has the same length as h_i. Accordingly, the Eq.(4) can be revised as follows:

$$G_i = tanh(W_g \tilde{h}_i + b_g) \tag{5}$$

where $W_g \in \mathbb{R}^{d \times d}$ is the weight matrix, b_g is the bias term. Then we apply the score vector G_i to integrate $\tilde{H}_{ori}$ and $\tilde{H}_{mask}$ to get the distinguishable features

C, which is calculated as follows:

$$C = \tilde{H}_{ori} + G \odot \tilde{H}_{mask} \tag{6}$$

Here, $\odot$ denotes the element-wise multiplication.

3.3 Interaction and Pooling Layer

To fully exploit the distinguishable features C, we devise an interaction layer to further compare each token in two sentences:

$$\hat{C} = TransformerBlock(C) \tag{7}$$

By performing interaction on the enhanced features C, the model can perceive the differentiated information of the sentence pair and enable better refinement of the features from both sequences. To obtain the high-level differentiation-aware representation for the sentence pair, we then employ an additional attention layer to aggregate the output $\hat{C} = \{\hat{c}_1, ..., \hat{c}_N\}$ of interaction layer:

$$\alpha_i = \frac{exp(w_a^T \hat{c}_i / \sqrt{d})}{\sum_{j=1}^{N} exp(w_a^T \hat{c}_j / \sqrt{d})} \tag{8}$$

$$h_{dif} = \sum_{i=1}^{N} \alpha_i \hat{c}_i \tag{9}$$

where $w_a \in \mathbb{R}^{d \times 1}$ is the trainable parameter. Then we feed the final output h_{dif} of this layer to the relation classifier module for the final prediction.

3.4 Relation Classifier

The semantic information of the original sequence S_{ori} is completely preserved in the global representation h_{cls}^{ori}, while h_{dif} is a differentiation-aware representation with attentive and distinguishable features. Combining them enables the model better to determine the semantic relation between the two sentences. Therefore, we concatenate them to make the final classification:

$$p(y|S_a, S_b) = FFN([h_{cls}^{ori}; h_{dif}]) \tag{10}$$

where $FFN(\cdot)$ is a feed forward network with one layer. During the training stage, the training object is to minimize the binary cross-entropy loss.

4 Experimentation

4.1 Corpora and Hyperparameter Settings

Corpora. We conduct experiments on three STM benchmarks: two English corpora QQP [9] and MRPC [5], and one Chinese corpus BQ [1]. Both QQP

Table 2: Statistics of three corpora QQP, MRPC and BQ. "Avg. words" denotes the average number of words of all sentences, and "Avg. shared words" is the average number of shared words of all sentence pairs.

Corpora	Size	Avg. words	Avg. shared words	Domain
QQP	404,276	11.06	10.08	open-domain
MRPC	5,801	21.89	31.39	open-domain
BQ	120,000	11.64	7.82	bank

and MRPC are open-domain corpora collected from online websites, while BQ is a domain-specific corpus derived from bank service logs. Each sentence pair in these corpora is associated with a binary label indicating whether they are the same in semantics. Data statistics are shown in Table 2, where we report the details of instances that contain repetitive content.

Hyperparameters. We use the pre-trained BERT and RoBERTa released by the huggingface community[1], and fine-tune them on each corpus. The hyperparameters are set as follows. We use AdamW [15] (β_1=0.9, β_2=0.999, ϵ = 1e-8) with weight decay of 0.01 to fine-tune the parameters. We set the initial learning rate to 2e-5 and decrease its value with linear scheduling as the model training. As for batch size, we use 64 for QQP and BQ, and 16 for MRPC. We fine-tune the model for five epochs and evaluate the model after every 200 steps. Checkpoints with the best performance on the development set are evaluated on the test set to report the performance. All of our experiments are conducted on a single Nvidia Tesla V100-16GB GPU.

4.2 Main Results

The main results are shown in Table 3 and Table 4. The reported results are average scores using five different seeds, and all the improvements over baselines are statistically significant ($p - value < 0.05$ in the statistical significance test).

English STM We compare our FDF to the previous STM models on QQP and MRPC. The results are shown in Table 3. All the models in Table 3 can be divided into three groups. The first group contains traditional neural matching models without pre-training, while the second and third groups comprise PLMs and the ones which employ PLMs as their backbones.

Table 3 shows that PLMs have salient performance gains over the traditional method due to pre-training on large-scale corpora. From the results of the MRPC corpus, we can observe that PLMs show their superiority, especially when performed on the small-scale corpus. It is worth mentioning that DRCN is capable of preserving both the original and the co-attentive feature information, while DRr-Net repeatedly reads the important words to understand the

[1] `https://huggingface.co/`

Table 3: Results (*Acc.*) on QQP and MRPC.

Models	**QQP**	**MRPC**
BiMPM [20]	88.2	-
DIIN [7]	89.1	-
DRCN [10]	90.2	82.5
DRr-Net [22]	89.8	82.9
BERT [4]	90.9	82.7
-large version	91.0	85.9
SS-BERT [13]	91.4	-
R^2-Net [23]	91.6	84.3
DC-Match [24]	91.2	83.8
FDF (BERT-base)	**91.6**	**84.8**
RoBERTa [14]	91.4	87.2
-large version	92.0	88.3
DC-Match (RoBERTa-large)	92.2	88.9
FDF (RoBERTa-large)	**92.4**	**89.3**

Table 4: Results on BQ.

Models	**BQ**	
	Acc.	F1
Text-CNN [8]	68.5	69.2
BiLSTM [17]	73.5	72.7
Lattice-CNN [12]	78.2	78.3
BiMPM [20]	81.9	81.7
ESIM [2]	81.9	81.9
LET [16]	83.2	83.0
BERT-wwm [3]	84.9	84.3
BERT-wwm-ext [3]	84.7	83.9
ERNIE [19]	84.7	84.2
BERT [4]	84.5	84.0
LET-BERT [16]	85.3	85.0
FDF (BERT)	**85.4**	**85.4**

sentences better. The performance of both DRCN and DRr-Net is close to that of BERT-base, which is impressive for models without pre-training.

The second group shows the performance of solely utilizing BERT [4], and the ones expanding BERT in different ways, such as the most representative R^2-Net. Benefiting from the pairwise relation learning processing, R^2-Net is able to make full use of the relation information. It achieves the best performance among the BERT-based models at the earlier time. By contrast, FDF outperforms R^2-Net on MRPC and obtains comparable performance on QQP.

For a better comparison, we also conduct experiments using RoBERTa-large [14] as baseline, and the results are shown in the third group in Table 3. Within the ones using RoBERTa-large as the backbone, DC-Match achieves the best performance on both MRPC and QQP corpora. Instead of performing text comparison by processing each word uniformly, DC-Match matches the intents and keywords under different levels of granularity. However, this approach could lead to incomplete semantic information, and thus may affect the model performance. In contrast, FDF is able to better preserve the semantics through the fusion and interaction layers. It can be observed that FDF outperforms DC-Match on the two corpora. All the results from Table 3 prove the necessity of reducing distractions by highlighting distinguishable features.

Chinese STM Furthermore, we evaluate FDF on the Chinese benchmark BQ, which is a domain-specific corpus for bank question matching. Following Lyu et al. [16], we also report the F1-score besides the accuracy rate (*Acc.*). Table 4 shows the comparison results of different models. Note that the models in Table 4 can be divided into two categories: BERT-free models and BERT-based models.

Within the models, LET-BERT uses word lattices and introduces HowNet's knowledge to solve word sense disambiguation. Therefore, it achieves impressive performance among all the models, including both BERT-free and BERT-based ones. Although both LET-BERT and FDF utilize graph neural networks to

Table 5: Ablation performance (*Acc.*) of FDF network.

Model	**QQP**	**MRPC**	**BQ**
FDF	**91.55**	**84.83**	**85.37**
w/o masking	91.35	84.16	84.80
simple gate	91.43	84.19	84.77
w/o fusion	91.43	83.13	84.70
w/o interaction	91.40	83.97	84.73
BERT-base	90.91	82.70	84.50

extract and represent features of local structures. Though, FDF does not utilize external information. Briefly, FDF is more straightforward but outperforms LET-BERT. The experimental results of FDF on BQ indicate that it performs effectively in different languages and domain-specific scenarios.

4.3 Ablation Study

We verify the possible contributions of different components in FDF by ablation study. The results are shown in Table 5. To validate the effectiveness of the masking strategy, we replace the masked sequence with the original sequence in Fig.1 and remain the rest of the FDF network unchanged to eliminate the effect of the number of parameters. It can be observed that the accuracy decreased by 0.2, 0.67 and 0.57 on QQP, MRPC and BQ respectively, when the shared-token masking is disabled. This demonstrates the effectiveness of exclusive feature extraction by deduplication. Further, we replace the multi-dimensional gate module with a simplified gate module. Specifically, we merely calculate a single scalar score to integrate two representations. It can be found that accuracy decreased to 91.43, 84.19 and 84.77 on the three datasets. This proves that treating each feature differently can obtain a better integration of representations.

Besides, when we remove the fusion layer and integrate the representations by directly adding them together, the performance also decreases. This implies that the fusion layer is helpful for producing distinguishable features. Moreover, the MRPC dataset possesses a longer text length (which can be observed from Table 2), thus making it more challenging for the STM model to determine whether the given pairs are semantically equivalent. In this case, the direct summation of the results from two channels cannot yield useful token representations for supporting the subsequent modules to predict the semantic relations. Therefore, the performance of FDF on the MRPC benchmark drops dramatically by 1.7 when the fusion layer is removed. In the last ablation, we remove the interaction layer and directly aggregate the outputs of the fusion layer. It can be observed that performance drops on all corpora. This demonstrates that the enhanced representations obtained by the fusion layer are not fully exploited, and the interaction layer enables further comparison between the sentence pair.

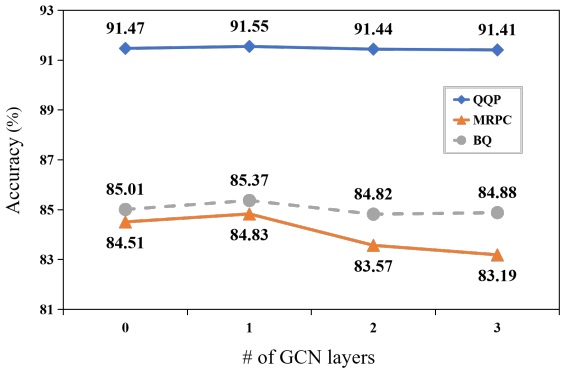
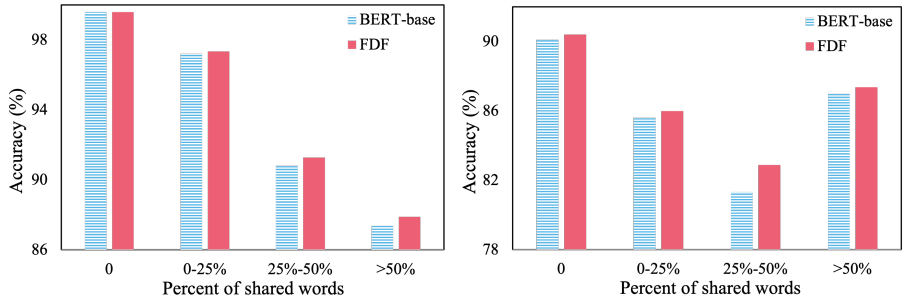


Fig. 3: Performance (*Acc.*) of FDF with different GCN layers.



(a) QQP (b) BQ

Fig. 4: Comparison between Baselines and FDFs on QQP and BQ with different proportions of shared words. The accuracy metric (*Acc.*) is considered.

4.4 Effect of GCN Layers

We validate the performance of FDF with different numbers of GCN layers. Fig. 3 shows the accuracy curves over three corpora. It can be found that FDF approaches the best performance on all corpora when we set the number to 1.

By contrast, both disabling GCN and utilizing more GCN layers lead to performance degradation. Frankly, when GCN is disabled, FDF cannot effectively model interactions between shareable and exclusive features. This most probably causes performance reduction. When we use a larger number of GCN layers, the model suffers from less relevant information or even the noises introduced by GCN, which can hamper the model from making final predictions.

Table 6: Examples of sentence pairs from the QQP and BQ corpora. Words in **bold** are the distinctions between the two sentences. **PLM** and **FDF** represent the prediction of the BERT-base and FDF, respectively.

ID	Sentence pair	Label	PLM	FDF
1	Who is the best singer **now**? Who is the best singer **of all time**?	0	1	0
2	What **should I do** to sleep better? What **is the best way** to sleep better?	1	0	1
3	**我**什么时候可以使用微利贷 (When can I use the loan app) 什么时候可以**再次**使用微粒贷 (When can I use the loan app **again**)	0	1	0

4.5 Effectiveness Analysis

To verify the effectiveness of FDF when it deals with the sentence pairs possessing different proportions of shared words, we split the validation sets into quarters and validate the performance of models trained on the original training set. Fig.4 shows the comparison of FDF and baseline on QQP and BQ.

It can be observed that when the proportion increases, the performance of the baseline usually decreases. As shown in Fig.4(a), BERT achieves nearly 100% accuracy when no shared word occurs in the sentence pair. Though, when the proportion is more than 50%, the accuracy drops dramatically. By contrast, FDF consistently yields improvements over the baseline when different proportions are considered. Specifically, when the proportion is higher than 25%, FDF achieves a substantial improvement compared to the baseline. When the proportion lies between 25% and 50%, FDF gains the improvements of 1.57% and 0.47% *Acc* on BQ and QQP, respectively. When the proportion increases to above 50%, improvements slightly reduce. The results imply that FDF can alleviate the distraction of repetitive contents effectively though incompletely, when the duplication is overly severe.

4.6 Case Study

This section presents several sample cases with predicted labels of FDF and the fine-tuned BERT in Table 6. These cases show that the original PLMs are confused by the words shared by both sentences and therefore fail to identify their semantic relations. For example, in the NO.1 and NO.3 cases, two sentences differ only in a few words (e.g., "*now*" and "*of all time*" in the No.1 case). The PLMs fail to highlight the non-repetitive contents of the pair and thus make the wrong prediction. By contrast, FDF is capable of producing distinguishable features to support the STM model to determine semantic consistency.

In addition, the baseline model is prone to classify sentence pairs that are partially different but contain the same meaning as negative cases. As shown in the NO.2 case, two sentences differ in "*should I do*" and "*is the best way*", the

baseline fails to identify their relation. During the training stage, FDF is able to highlight the representations of these phrases and employ the ground-truth labels to guide the model to learn that these phrases actually express the same meaning in this scenario.

5 Conclusion

We propose a Feature Differentiation and Fusion network (FDF) to enhance the current PLMs-based STM models. It is designed to alleviate the distraction of repetitive contents between sentences, conducting separate extraction of shareable and exclusive features and gated information fusion by GCN. Experiments on the benchmark corpora demonstrate the effectiveness of FDF.

In the future, we will carry out the study of the general multilingual FDF networks for STM. It is motivated by the findings in this paper that some linguistic features and commonsense knowledge are shareable among different languages for signaling semantic consistency. This implies the possibility of utilizing the Parent-Child learning model to transform experiences (parameters of neurons) across different languages. Progressive and contrastive learning will be used.

Acknowledgements We thank all reviewers for their insightful comments, as well as the great efforts our colleagues have made so far. This work is supported by National Key R&D Program of China (2020YFB1313601) and National Science Foundation of China (62076174, 62076175).

References

1. Chen, J., Chen, Q., Liu, X., Yang, H., Lu, D., Tang, B.: The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4946–4951
2. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: 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
3. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z.: Pre-training with whole word masking for chinese BERT. IEEE ACM Trans. Audio Speech Lang. Process. **29**, 3504–3514
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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
5. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the Third International Workshop on Paraphrasing (IWP2005)
6. Gao, J., He, D., Tan, X., Qin, T., Wang, L., Liu, T.: Representation degeneration problem in training natural language generation models. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019

7. Gong, Y., Luo, H., Zhang, J.: Natural language inference over interaction space. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings
8. He, T., Huang, W., Qiao, Y., Yao, J.: Text-attentional convolutional neural network for scene text detection. IEEE Trans. Image Process. **25**(6), 2529–2541
9. Iyer, S., Dandekar, N., Csernai, K.: First quora dataset release: Question pairs (2017), `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`
10. Kim, S., Kang, I., Kwak, N.: Semantic sentence matching with densely-connected recurrent and co-attentive information. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 6586–6593
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings
12. Lai, Y., Feng, Y., Yu, X., Wang, Z., Xu, K., Zhao, D.: Lattice cnns for matching based chinese question answering. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 6634–6641
13. Liu, T., Wang, X., Lv, C., Zhen, R., Fu, G.: Sentence matching with syntax- and semantics-aware BERT. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 3302–3312
14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019
16. Lyu, B., Chen, L., Zhu, S., Yu, K.: LET: linguistic knowledge enhanced graph transformer for chinese short text matching. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021. pp. 13498–13506
17. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 2786–2792
18. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: Disan: Directional self-attention network for rnn/cnn-free language understanding. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, New Orleans, Louisiana, USA, February 2-7, 2018. pp. 5446–5455
19. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223
20. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. pp. 4144–4150
21. Xia, T., Wang, Y., Tian, Y., Chang, Y.: Using prior knowledge to guide bert's attention in semantic textual matching tasks. In: WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 2466–2475

22. Zhang, K., Lv, G., Wang, L., Wu, L., Chen, E., Wu, F., Xie, X.: Drr-net: Dynamic re-read network for sentence semantic matching. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 7442–7449
23. Zhang, K., Wu, L., Lv, G., Wang, M., Chen, E., Ruan, S.: Making the relation matters: Relation of relation learning network for sentence semantic matching. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021. pp. 14411–14419
24. Zou, Y., Liu, H., Gui, T., Wang, J., Zhang, Q., Tang, M., Li, H., Wang, D.: Divide and conquer: Text semantic matching with disentangled keywords and intents. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 3622–3632