Divide and Conquer: Text Semantic Matching with Disentangled Keywords and Intents

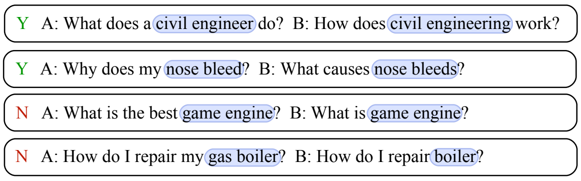
Yicheng Zou[1,2], **Hongwei Liu**[2], **Tao Gui**[1*], **Junzhe Wang**[2], **Qi Zhang**[2*],
Meng Tang[3], **Haixiang Li**[3], **Daniel Wang**[3]
[1]Institute of Modern Languages and Linguistics, Fudan University, Shanghai, China
[2]School of Computer Science, Fudan University, Shanghai, China
[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing, Shanghai, China
[3]IPS, Tencent PCG, Beijing, China
{yczou18,tgui,qz}@fudan.edu.cn, daniellwang@tencent.com

Abstract

Text semantic matching is a fundamental task that has been widely used in various scenarios, such as community question answering, information retrieval, and recommendation. Most state-of-the-art matching models, e.g., BERT, directly perform text comparison by processing each word uniformly. However, a query sentence generally comprises content that calls for different levels of matching granularity. Specifically, *keywords* represent factual information such as action, entity, and event that should be strictly matched, while *intents* convey abstract concepts and ideas that can be paraphrased into various expressions. In this work, we propose a simple yet effective training strategy for text semantic matching in a divide-and-conquer manner by disentangling keywords from intents. Our approach can be easily combined with pre-trained language models (PLM) without influencing their inference efficiency, achieving stable performance improvements against a wide range of PLMs on three benchmarks.

1 Introduction

Text semantic matching aims to predict a matching category or a matching score reflecting the semantic similarity given a pair of text sequences, which is a fundamental task employed in a wide range of applications (Huang et al., 2013; Hu et al., 2014; Palangi et al., 2016; Cer et al., 2017; Rücklé et al., 2020; Pang et al., 2021). Recently, pre-trained language models (PLM) show remarkable capability of representation learning and have accelerated the research of text semantic matching (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019). They typically exploit large-scale corpora and well-designed self-supervised learning objectives to better learn semantic representations, achieving state-of-the-art performances or even surpassing the level of non-expert humans on general semantic matching benchmarks (Wang et al., 2019b,a).



Y A: What does a civil engineer do? B: How does civil engineering work?

Y A: Why does my nose bleed? B: What causes nose bleeds?

N A: What is the best game engine? B: What is game engine?

N A: How do I repair my gas boiler? B: How do I repair boiler?

Figure 1: Examples of sentence pairs sampled from the QQP dataset. The **keywords** are highlighted, while the other words constitute abstract **intents**. **Y** and **N** represent whether the pair is matched or not. The original matching problem can be decomposed into two subproblems: keyword matching and intent matching. A semantically equivalent pair generally means the keyword and intent are matched simultaneously.

Most existing PLMs aim to establish a foundation for various downstream tasks (Bommasani et al., 2021) and focus on finding a generic way to encode text sequences. When applied to the task of text semantic matching, it is a common practice to add a simple classification objective for fine-tuning and directly perform text comparison by processing each word uniformly. Nevertheless, each sentence can be typically decomposed into content with different levels of matching granularity (Su et al., 2021). Exemplar sentence pairs can be found in Figure 1. The primary content refers to *keywords* that reflect the factual information about entities or actions, which should be strictly matched. The other content constitute abstract *intents*, which can be generally paraphrased into various expressions to convey the same concepts or ideas.

Considering the situation where sentence content has different levels of matching granularity, we propose **DC-Match**, a simple but effective training regime for text semantic matching in a divide-and-conquer manner. Specifically, we break down the matching problem into two sub-problems: *keyword matching* and *intent matching*. Given a pair of input text sequences, the model learns to disentangle keywords from intents by utilizing the method of distant supervision. In addition to the standard se-

[*]Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3622 - 3632
May 22-27, 2022 ©2022 Association for Computational Linguistics

quence matching that has a global receptive field, we further match keywords and intents separately to learn the way of content matching under different levels of granularity. Finally, we design a special training objective that combines the solutions to the sub-problems, which minimizes the KL-divergence between the global matching distribution (original problem) and the joint keyword-intent matching distribution (sub-problems). At inference time, we expect that the global matching model automatically distinguishes keywords from intents, then makes final predictions conditioned on the disentangled content in different matching levels.

We adopted DC-Match to a wide range of PLMs. Comprehensive experiments were conducted on two English text matching benchmarks MRPC (Dolan and Brockett, 2005) and QQP (Iyer et al., 2017), and a Chinese benchmark Medical-SM. Our approach can be easily combined with PLMs plus few additional parameters, but still achieves stable performance improvements against most baseline PLMs. Notably, all the auxiliary procedures and parameters are only involved in the training stage. The inference efficiency of our approach is exactly the same as that of PLM baselines, without additional parameters and computations. Our codes and datasets are publicly available[1].

Our contributions are three-fold: 1) We introduce a novel training regime for text matching, which disentangles keywords from intents based on different levels of matching granularity in a divide-and-conquer manner. 2) The proposed approach is simple yet effective, which can be easily combined with PLMs plus few auxiliary training parameters while not changing their original inference efficiency. 3) Experimental results on three benchmarks across two languages demonstrate the effectiveness of our approach in different aspects.

2 Related Work

Text semantic matching plays an important role in many applications, such as Information Retrieval (IR) and Natural Language Inference (NLI). Traditional technologies exploit neural networks with different inductive biases, e.g., CNN (Tan et al., 2016), RNN (Tai et al., 2015; Cheng et al., 2016), GNN (Wu et al., 2020), and attention mechanism (Parikh et al., 2016; Chen et al., 2017). To enhance the matching performance, dozens of works use richer syntactic or hand-crafted features (Chen et al., 2017; Tay et al., 2018b; Gong et al., 2018; Kim et al., 2019), add complex alignment computations (Wang et al., 2017; Tan et al., 2018; Gong et al., 2018; Yang et al., 2019), and perform multi-pass matching procedures (Tay et al., 2018a; Kim et al., 2019), which shows the effectiveness of representation-oriented approaches and model designing strategies based on information interaction.

Recently, large-scale pre-trained language models (PLM) have boosted the performance of text semantic matching by making full use of massive text resources. Most of them are composed of multiple transformer layers (Vaswani et al., 2017) with multi-head attentions and are pre-trained with well-designed self-supervised learning objectives. Representative models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019) aim to establish a powerful encoder that has a comprehensive understanding of input texts. For the task of text semantic matching, PLMs can be fine-tuned under a paradigm of sequence classification with only an additional classification layer, achieving state-of-the-art performances on general semantic matching benchmarks (Wang et al., 2019b,a). PLMs can be regarded as foundation models (Bommasani et al., 2021) and they mainly focus on finding a generic way to encode text sequences. Instead of processing each word uniformly, in this work, we devise a novel training regime that processes sentence pairs by disentangling keywords from intents, which can be easily combined with PLMs to stack additional improvements to text semantic matching.

3 Methodology

In this section, we detail the proposed training regime DC-Match. It consists of three training objectives: a classification loss for the global matching model; a distantly supervised classification loss that learns to distinguish keywords from intents; a special training objective following the idea of divide and conquer, which uses the KL-divergence to ensure that the global matching distribution (original problem) is similar to the distribution of combined solutions to disentangled keywords and intents (sub-problems). The overall framework is illustrated in Figure 2.

3.1 Text Semantic Matching using PLMs

First, we formally define the task of text semantic matching and describe a generic way for this task

[1] https://github.com/RowitZou/DC-Match

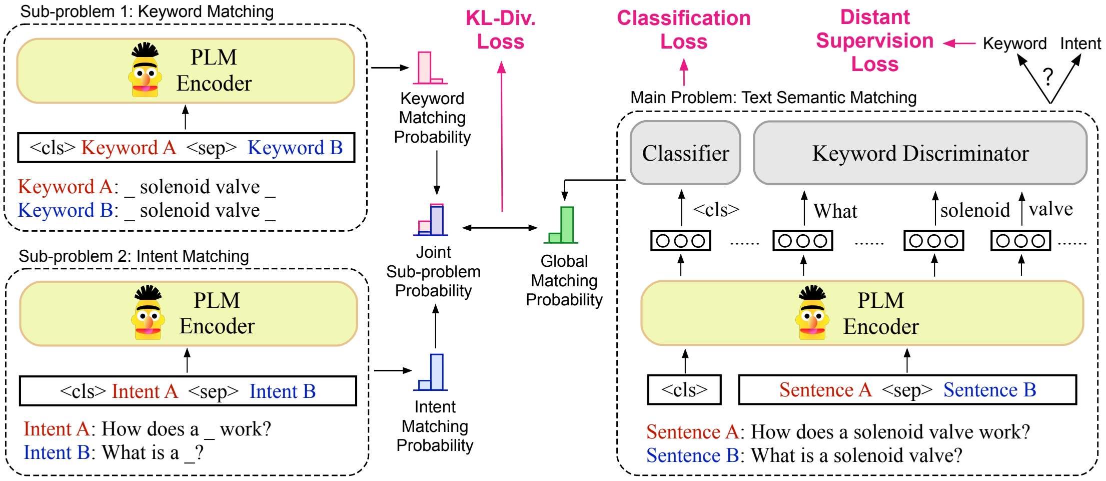

Sub-problem 1: Keyword Matching
PLM Encoder
<cls> Keyword A <sep> Keyword B
Keyword A: _ solenoid valve _
Keyword B: _ solenoid valve _
Sub-problem 2: Intent Matching
PLM Encoder
<cls> Intent A <sep> Intent B
Intent A: How does a _ work?
Intent B: What is a _?
KL-Div. Loss
Keyword Matching Probability
Joint Sub-problem Probability
Intent Matching Probability
Global Matching Probability
Classification Loss
Distant Supervision Loss
Keyword Intent
?
Main Problem: Text Semantic Matching
Classifier
Keyword Discriminator
<cls>
What
solenoid
valve
PLM Encoder
<cls>
Sentence A <sep> Sentence B
Sentence A: How does a solenoid valve work?
Sentence B: What is a solenoid valve?

Figure 2: Overview of DC-Match. The training regime has three objectives: (1) a standard matching classification loss; (2) a distant supervision loss for keyword and intent discrimination; (3) a KL-divergence loss that makes the global matching probability (main problem) consistent with the probability of combined solutions to keyword matching and intent matching (sub-problems).

by using PLMs. Given two text sequences $S^a = \{w_1^a, w_2^a, ..., w_{l_a}^a\}$ and $S^b = \{w_1^b, w_2^b, ..., w_{l_b}^b\}$, the goal of text semantic matching is to learn a classifier $y = \xi(S^a, S^b)$ to predict whether S^a and S^b is semantically equivalent. Here, w_i^a and w_j^b represent the i-th and j-th word in the sequences, respectively, and l_a, l_b denote the sequence length. y can be either a binary classification target indicating whether or not the two sequences are matched, or a multi-class classification target that reflects different matching degrees.

Recently, pre-trained language models (PLM) have achieved remarkable success in text understanding and representation learning (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019). They are pre-trained on large-scale text corpora with heuristic self-supervised learning objectives, and can be served as a powerful sequence classifier by fine-tuning on the downstream classification task. For text semantic matching, it is a common practice that we directly concatenate S^a and S^b as a consecutive sequence $S^{a,b} = [S^a; w^{sep}; S^b]$ by a separator token w^{sep} and feed it into the PLM encoder:

$$[\mathbf{h}^{cls}; \mathbf{H}^{a,b}] = \mathrm{PLM}([w^{cls}; S^{a,b}]), \tag{1}$$

$$P(y|S^a, S^b) = \mathrm{Softmax}(\mathbf{h}^{cls} \cdot \mathbf{W}^\top). \tag{2}$$

Here, w^{cls} is a special token in front of each sequence, and the final hidden state corresponding to this token $\mathbf{h}^{cls}$ is used as the aggregate sequence representation. During fine-tuning, only a single classification layer is introduced to make the final prediction, where $\mathbf{W} \in \mathbb{R}^{K \times H}$ represents trainable weights and K is the number of labels. Finally, we compute a standard classification loss for fine-tuning as follows:

$$\mathcal{L}_{sm} = -\log P(y|S^a, S^b). \tag{3}$$

3.2 Disentangling Keyword from Intent with Distant Supervision

Most existing PLMs aim to find a generic way to encode text sequences and establish a foundation for language understanding. For different classification tasks, e.g., sentiment analysis, text semantic matching, and natural language inference, the PLM typically exploits the same fine-tuning paradigm, and processes text sequences in a straightforward and uniform way. In this work, inspired by previous works of decomposable paraphrase generation (Li et al., 2019; Su et al., 2021), we introduce a task-specific assumption to the text semantic matching, and postulate that each sentence could be decomposed into keywords and intents. Intuitively, keywords represent factual information such as actions and entities that should be strictly matched, while intents convey abstract concepts or ideas that can be expressed in different ways. By disentangling keywords from intents, the matching procedure can be divided into two easier sub-problems that call for different levels of matching granularity.

However, automatic disentanglement of keywords and intents is not easy due to the lack of manually annotated data. To address this problem, following recent research on distant supervision (Liang et al., 2020; Meng et al., 2021), we use a rule-based method to automatically generate keyword labels by extracting entity mentions in the raw text based on the entities in external knowledge bases (see details in Section 4.2). All extracted entities are labeled as keywords and the remainder of the sentence words are labeled as intents. After obtaining the weakly labeled information, we add an auxiliary training objective that forces the model to learn disentangled keyword and intent representations. Formally, given the output states $\mathbf{H}^{a,b}$ from PLM in Eq.1, we split the states into two groups $\mathbf{H}_k^{a,b} \in \mathbb{R}^{N_k \times H}$ and $\mathbf{H}_i^{a,b} \in \mathbb{R}^{N_i \times H}$ that correspond to the tokens of keywords and intents, respectively, where N_k, N_i denote the token number. Then, the keyword-intent classification loss is defined as follows:

$$\mathcal{L}_{ds} = -[\log\sigma(\hat{\mathbf{h}}_k^{a,b}\mathbf{W}_{ds}^{\top}) + \log\sigma(-\hat{\mathbf{h}}_i^{a,b}\mathbf{W}_{ds}^{\top})], \tag{4}$$

where $\mathbf{W}_{ds} \in \mathbb{R}^{1\times H}$ is trainable parameters, and $\hat{\mathbf{h}}_k^{a,b}, \hat{\mathbf{h}}_i^{a,b}$ are transformed by $\mathbf{H}_k^{a,b}, \mathbf{H}_i^{a,b}$ using average pooling. The objective in Eq.4 aims to push the encoder to learn representations of keywords and intents such that they are far apart from each other, modeling disentangled sentence content in different matching levels.

3.3 Divide-and-Conquer Matching Strategy

The auxiliary training objective in Eq.4, nevertheless, cannot be directly associated with the original text matching problem. To facilitate the true contributions of keywords and intents to the final prediction, we introduce a novel matching strategy following the idea of divide and conquer. Specifically, we divide the original matching problem into two easier sub-problems: keyword matching and intent matching, and assume that they are independent to each other. The solutions to the sub-problems are then combined to give a solution to the original problem. Recall that the goal of text semantic matching is to learn $y = \xi(S^a, S^b)$ where y can be either a binary classification target or a multi-class classification target. We assume that each sub-problem follows the same type of target, and the probability distribution of combined solutions $Q(y)$ can be derived from the joint probability distribution of the two sub-problems $P(y_k, y_i)$ as:

$$\begin{aligned} Q(y = c_n) = \; & P(y_k = c_n, y_i = c_n) \\ & + \textstyle\sum_{c_m > c_n} P(y_k = c_n, y_i = c_m) \\ & + \textstyle\sum_{c_m > c_n} P(y_k = c_m, y_i = c_n). \end{aligned} \tag{5}$$

Here, c_n, c_m denote the target classes which reflect the matching degrees, and $c_m > c_n$ means c_m has a higher matching score than c_n. For example, in a three-class scenario, $y \in \{2, 1, 0\}$ means exact match, partial match, and mismatch, respectively, and $Q(y = 0)$ is the probability that at least one of the sub-problems is inferred as mismatched.

To model the sub-problems, we reuse the matching model in Eq.1 and Eq.2 to separately compare keywords and intents and get conditional probabilities $P(y_k|S_k^a, S_k^b)$ and $P(y_i|S_i^a, S_i^b)$. S_k and S_i represent text sequences where tokens of intents or keywords are masked, respectively. Then, under the assumption of independent sub-problems, the conditional joint distribution of y_k and y_i is:

$$P(y_k, y_i|S^a, S^b) = P(y_k|S_k^a, S_k^b)P(y_i|S_i^a, S_i^b). \tag{6}$$

Finally, we can combine the solutions to the sub-problems and compute the conditional distribution $Q(y|S^a, S^b)$ using Eq.5. To ensure that the global matching distribution (original problem) is similar to the distribution of combined solutions to sub-problems, we use the bidirectional KL-divergence loss to minimize the distance between $P(y|S^a, S^b)$ and $Q(y|S^a, S^b)$ as follows:

$$\begin{aligned} \mathcal{L}_{dc} = 1/2 \cdot (& D_{KL}[P(y|S^a, S^b)||Q(y|S^a, S^b)] \\ & + D_{KL}[Q(y|S^a, S^b)||P(y|S^a, S^b)]). \end{aligned} \tag{7}$$

By this means, we expect that the global matching model learns to make final predictions with better interpretability, which are conditioned on the disentangled keywords and intents that require different levels of matching granularity.

3.4 Training and Inference

At the training stage, we combine the three loss functions $\mathcal{L}_{sm}, \mathcal{L}_{ds}, \mathcal{L}_{dc}$ to jointly train the model:

$$\mathcal{L} = \mathcal{L}_{sm} + \mathcal{L}_{ds} + \mathcal{L}_{dc}. \tag{8}$$

At the inference time, we directly infer the matching category for a sentence pair based on the conditional probability of the original problem, namely $y^* = \mathrm{argmax}_y P(y|S^a, S^b)$. It means our inference procedure is exactly the same as that of PLM baselines without additional computations. Here, we do not infer matching results from the probability of combined solutions $Q(y|S^a, S^b)$, since annotation information of keywords and intents is generally not available at the inference time,

Split	# of pairs	Avg. length	# of pairs in categories		
			EM(2)	PM(1)	MM(0)
Train	38,406	12.25	7,754	18,617	12,035
Dev.	4,801	12.25	975	2,329	1,497
Test	4,801	12.19	938	2,315	1,548

Table 1: Statistics of the Medical-SM dataset. Each query pair can be categorized into exact match (EM), partial match (PM), or mismatch (MM).

	QQP	MRPC	Medical
# keywords in each pair	2.38	6.53	2.51
# tokens in each keyword	1.98	1.68	4.51
BLEU (match)	.1451	.3088	.2754
BLEU (mismatch)	.0961	.2155	.1284

Table 2: Statistics of distantly labeled keywords on training sets. BLEU (match/mismatch) denotes the keyword BLEU score in matched/mismatched pairs, respectively.

and $Q(y|S^a, S^b)$ cannot be directly computed. Although we use external corpora to automatically obtain distant labels, it might induce incomplete and noisy signals (Meng et al., 2021), introducing biases to $Q(y|S^a, S^b)$ approximation. Hence, we only use distant labels at the training stage as auxiliary information augmentation to the global matching model. Nevertheless, we observe that after model training, $P(y|S^a, S^b)$ is highly consistent with $Q(y|S^a, S^b)$ (see details in Section 5.4). As a result, a high-quality set of keyword labels might bring additional performance enhancement by better approximating $Q(y|S^a, S^b)$.

4 Experimental Settings

4.1 Datasets

We evaluate our approach and all baselines on three benchmarks for text semantic matching: two English datasets MRPC (Dolan and Brockett, 2005) and QQP (Iyer et al., 2017), and one Chinese dataset Medical-SM. Both MRPC and QQP are corpora of sentence pairs automatically extracted from online websites, with annotated binary classification labels indicating whether the sentences in the pair are semantically equivalent. We use the official dataset collections on Glue (Wang et al., 2019b) released by the community[2], where MRPC contains 5,801 sentence pairs and QQP consists of 404,276 annotated sentence pairs[3].

Furthermore, we evaluate our approach on a Chinese text matching dataset Medical-SM, which consists of user-generated query pairs collected from a Chinese search engine. The dataset contains 48,008 query pairs in the domain of medical consulting. Each query pair can be categorized into three classes: exact match, partial match, or mismatch. The annotation is completed by five independent experts and we keep the labeling choices that most annotators accept. Statistics of our constructed dataset are shown in Table 1. To facilitate the research, we will release the dataset publicly.

4.2 Automatic Keyword Labeling

In this work, we generate distant supervision labels for identification of keywords and intents using a heuristic approach. Inspired by previous works for distantly supervised NER (Liang et al., 2020; Meng et al., 2021), we first extract potential keywords with part-of-speech tags of nouns, verbs, and adjectives obtained from NLTK (Bird, 2006). We then match these potential keywords by using external knowledge bases: wikipedia entity graph (Bhatia and Vishwakarma, 2018) for English corpora, and Sogou knowledge graph (Wang et al., 2019c) for Chinese Medical-SM. Finally, we use the binary IO format to label whether a token belongs to keywords or intents (Peng et al., 2019). Table 2 shows the statistics of distantly labeled keywords on the training sets of three benchmarks. We use BLEU score (Papineni et al., 2002) to measure the relevance of keywords between two compared sentences for both matched pairs and mismatched pairs. We observe that matched sentence pairs generally contain keywords with higher relevance. As a result, generic models might wrongly output high matching scores just conditioned on matched keywords regardless of their context, because models tend to learn statistical biases in the data (Manjunatha et al., 2019; Lin et al., 2021).

4.3 Implementation Details

For a fair comparison, we fine-tune each PLM of the original version and its DC-Match variant with the same set of hyper-parameters. The fine-tuning process of the QQP and MRPC datasets follows Wang et al. (2021). Specifically, we apply AdamW (Loshchilov and Hutter, 2018) (β_1=0.9, β_2=0.999) with a weight decay rate of 0.01 and set the learning rate to 2e-5. The batch size is set to 64 for QQP and 16 for MRPC. All experiments are con-

[2] https://huggingface.co/datasets/glue

[3] Since the labels for the official QQP test set are not released, we report evaluation results on the validation set.

Model	**QQP**	**MRPC**
CENN (Zhang et al., 2017)	80.7	76.4
L.D.C (Wang et al., 2016)	85.6	78.4
BiMPM (Wang et al., 2017)	88.2	-
DIIN (Gong et al., 2018)	89.1	-
DRCN (Kim et al., 2019)	90.2	82.5
DRr-Net (Zhang et al., 2019)	89.8	82.9
R^2-Net (Zhang et al., 2021)	91.6	84.3
BERT (Devlin et al., 2019)	90.9	82.7
-large version	91.0	85.9
RoBERTa (Liu et al., 2019)	91.4	87.2
-large version	92.0	87.6
ALBERT (Lan et al., 2019)	90.4	86.0
-large version	90.9	86.5
DeBERTa (He et al., 2020)	91.7	88.4
-large version	92.1	88.6
FunnelTF (Dai et al., 2020)	91.9	87.1
DC-Match (RoBERTa-base)	91.7	88.1
DC-Match (RoBERTa-large)	**92.2**	**88.9**

Table 3: Experimental results (**Accuracy**) on the **QQP** and **MRPC** text semantic matching datasets.

Model	**QQP**	**MRPC**
	Ori. → DC (change)	Ori. → DC (change)
BERT	90.91 → 91.16 (**0.25**)	82.66 → 83.82 (**1.16**)
-large	90.98 → 91.45 (**0.47**)	85.85 → 86.08 (0.23)
RoBERTa	91.41 → 91.69 (**0.28**)	87.24 → 88.05 (**0.81**)
-large	92.03 → 92.20 (**0.17**)	87.59 → 88.92 (**1.33**)
ALBERT	90.37 → 90.62 (**0.25**)	86.02 → 86.26 (0.24)
-large	90.91 → 90.94 (0.03)	86.49 → 87.01 (**0.52**)
DeBERTa	91.68 → 91.78 (0.10)	88.40 → 88.81 (**0.41**)
-large	92.13 → 92.22 (0.09)	88.57 → 89.21 (**0.64**)
FunnelTF	91.92 → 92.09 (**0.17**)	87.07 → 87.53 (**0.46**)

Table 4: Experimental results of **Accuracy** on the **QQP** and **MRPC** datasets. We compare the results of original PLMs with those using our DC-Match training strategy (Ori.→DC), and calculate the improvement of accuracy. Numbers in **bold** indicate whether the change is significant (using a Wilcoxon signed-rank test; $p < 0.05$).

ducted on a single RTX 3090 GPU. For QQP, we fine-tune the model for 50,000 steps and model checkpoints are evaluated every 2,000 steps. For MRPC, we fine-tune the model for 20 epochs and evaluate the model after each epoch. Checkpoints with top-3 performance on the development set are evaluated on the test set to report average results. For Medical-SM, we use the same fine-tuning strategy as for QQP, and use the chinese version of PLM checkpoints released by Cui et al. (2021)[4].

5 Results and Analysis

5.1 Main Results

Table 3 shows the main results of comparison models on the QQP and MRPC dataset. Following previous works (Zhang et al., 2021; Wang et al., 2021), we evaluate matching performance using **Accuracy** and some results are from their reported scores. In Table 3, all baselines are categorized into two groups. The first group includes traditional methods that exploit neural networks with different inductive biases, and the second group includes PLMs that benefit from large-scale external pre-training data. Unsurprisingly, PLMs show a superior performance against traditional neural matching models, especially on the small-scale dataset MRPC. When equipped with the DC-Match training strategy, PLMs can achieve further performance enhancement. In Table 3, we report the results of DC-Match that uses RoBERTa as the backbone PLM, which outperforms all baselines on both datasets. However, the improvement on a single PLM does not necessarily mean the effect of DC-Match has generalizability. Hence, to probe the effectiveness of our proposed training regime, we apply DC-Match to all the PLMs in the second group and report the results of performance change in Table 4. Notably, the listed PLMs generally have different architectures and parameter scales, and we fine-tune each PLM of the original version and its DC-Match variant using the same set of configurations without additional tuning of hyper-parameters. We are surprised to find that the matching accuracy of all PLMs increases stably on both datasets. It indicates that the divide-and-conquer strategy by breaking down the matching problem into easier sub-problems can effectively give a better solution to the original problem. Besides, from Table 4 we observe that DC-Match brings more significant performance boost to the small dataset MRPC, which probes that the information of keywords and intents is an important feature for text semantic matching, especially when the training data is too limited to find useful latent patterns.

Furthermore, we evaluate DC-Match on the Chinese Medical-SM. Different from QQP and MRPC, Medical-SM is a three-class classification dataset. In addition to accuracy, we further employ Macro-F1 to assess the quality of problems with multiple classes. From Table 5 we observe that DC-Match still boosts the matching performance of PLMs, indicating that our strategy works fine in a multi-class classification scenario and in different languages.

[4] Since the large version of Chinese BERT is not available, we use Chinese MacBERT (Cui et al., 2020) instead of BERT.

Model	Accuracy Ori. → DC (change)	Macro-F1 Ori. → DC (change)
BERT	73.55 → 73.83 (**0.28**)	72.91 → 73.15 (**0.24**)
-large	74.55 → 74.69 (0.14)	74.01 → 74.13 (0.12)
RoBERTa	73.19 → 73.73 (**0.54**)	72.43 → 72.96 (**0.53**)
-large	73.51 → 74.22 (**0.71**)	72.83 → 73.67 (**0.84**)

Table 5: **Accuracy** and **Macro-F1** on the **Medical-SM** corpus. Numbers in **bold** indicate the result change is significant (Wilcoxon signed-rank test; $p < 0.05$).

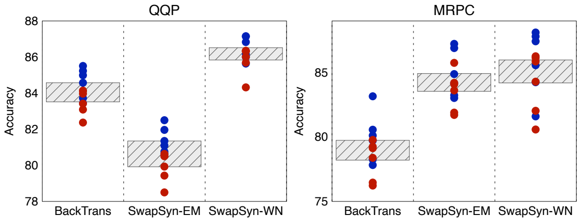


Figure 3: Robustness evaluation on the QQP and MRPC datasets. The x-axis denotes different text transformations that aim to test whether models are vulnerable to attacks. The y-axis denotes model accuracy on the transformed test set. Red dots represent the original PLMs while Blue dots represent those using the DC-Match strategy. Bar plots denote the gap of mean accuracy between two groups of models.

5.2 Ablation Experiments

We also perform ablation studies to validate the effectiveness of each part in DC-Match. Table 6 demonstrates the results of different settings for the proposed training strategy equipped with RoBERTa. After only adding the distantly supervised loss for keyword and intent identification (+$\mathcal{L}_{ds}$), we find that the results are not significantly different from the original PLMs. It reflects that this auxiliary training objective cannot be directly associated with the original text matching problem, so $\mathcal{L}_{ds}$ itself might not be helpful for the final target. However, if we remove $\mathcal{L}_{ds}$ from DC-Match and only keep the divide-and-conquer training objective (+$\mathcal{L}_{dc}$), we observe a performance degradation compared with the full version of DC-Match. It indicates that the distant supervision target helps the model learn to disentangle keywords from intents and obtain distinguished content representations that call for different levels of matching granularity, which might contribute to the solutions to sub-problems. Besides, the incorporation of the divide-and-conquer objective (both +$\mathcal{L}_{dc}$ and +$\mathcal{L}_{ds}, \mathcal{L}_{dc}$) improves the performance of PLMs to varying degrees, which manifests the effectiveness of the matching strategy in a decomposed manner.

Models	QQP	MRPC	Medical-SM
RoBERTa-base	91.41	87.24	73.19
+ $\mathcal{L}_{ds}$	91.48	87.36	73.30
+ $\mathcal{L}_{dc}$	91.61	87.88	73.65
+ $\mathcal{L}_{ds}, \mathcal{L}_{dc}$ (ours)	91.69	88.05	73.73
RoBERTa-large	92.03	87.59	73.51
+ $\mathcal{L}_{ds}$	91.96	87.86	73.85
+ $\mathcal{L}_{dc}$	92.15	88.82	74.13
+ $\mathcal{L}_{ds}, \mathcal{L}_{dc}$ (ours)	92.20	88.92	74.22

Table 6: Ablation study of DC-Match on three text semantic matching datasets. We report results of **Accuracy** and use RoBERTa as the backbone model.

5.3 Robustness Evaluation

The divide-and-conquer strategy disentangles keywords from intents, which provides additional interpretability for final matching judgements. Following Wang et al. (2021), we conduct robustness evaluation to probe whether DC-Match is robust to text transformations by breaking down the matching problem into easier sub-problems. Specifically, we use a public toolkit[5] and test the following transformations: (1) **BackTrans** transforms each sentence into a semantically equal sentence using back translation. (2) **SwapSyn-WN** replaces words with synonyms provided by WordNet (Miller, 1995). (3) **SwapSyn-EM** replaces common words with synonyms using Glove Embeddings (Pennington et al., 2014). We test 6 PLMs (BERT, ALBERT, RoBERTa with base and large version) in their original and DC-Match enhanced version, and report the results in Figure 3[6]. We observe that both original PLMs and their DC-Match variants suffer performance degradation. However, the DC-Match enhanced PLMs can keep a more stable performance compared to original ones, which manifests that DC-Match can improve the robustness of PLMs to a certain extent for the text semantic matching task.

5.4 Analysis of Divide-and-Conquer Strategy

Recall that the model cannot access the labeled keywords at test time, so the probability of combined solutions to the sub-problems $Q(y)$ cannot be directly computed. Hence, the KL-divergence loss in Eq.7 is designed to minimize the distance between $Q(y)$ and the global matching probability $P(y)$, aiming to simulate the divide-and-conquer process

[5] `https://www.textflint.io`

[6] All transformations are conducted on the subset of the original evaluation set where both the original PLMs and the DC-Match enhanced variants give accurate predictions.

Sentence Pair	Label	PLM	DC	Kw.	In.
A: What is the difference between an animal cell and a plant cell? B: What is the difference between plant cell vacuoles and animal cell vacuoles?	0	1	0	0	1
A: Benchmark Treasury 10-year notes gained 17/32, yielding 4.015 percent. B: The benchmark 10-year note was recently down 17/32, to yield 4.067 percent.	0	1	0	1	0
A: Is there any culture difference between US and UK? B: What is the biggest difference in British culture and American culture?	1	0	1	1	1
A: But the cancer society said its study had been misused. B: The American Cancer Society said the study was flawed in several ways.	0	1	0	0	0

Table 7: Test cases on the QQP and MRPC datasets. We use BERT-base as the backbone model. Words in Red represent distantly labeled keywords. **PLM**, **DC**, **Kw.**, and **In.** represent predictions from the original PLMs, the DC-Match enhanced PLMs, and the DC-Match sub-problems (keyword matching and intent matching), respectively.

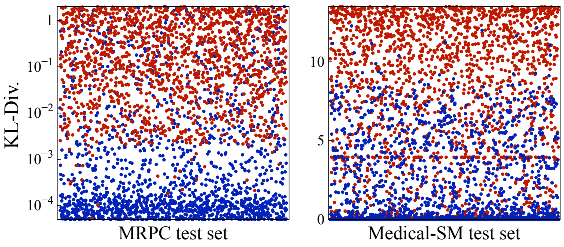


Figure 4: KL-divergence between $P(y)$ and $Q(y)$. Each point denotes the KL-divergence score of a test sample (1725 samples for MRPC and 4801 samples for Medical-SM). Red dots are scores from the original PLMs, while Blue dots are those from DC-Match. BERT-base is used as the backbone model. We observe that DC-Match significantly narrows the gap between $P(y)$ and $Q(y)$ compared to the original PLMs.

at inference time. To probe that $P(y)$ can truly approximate $Q(y)$, we further label the keywords in test sets as described in Section 4.2, so that we can calculate $Q(y)$ directly[7]. We compute the KL-divergence score between $P(y)$ and $Q(y)$ for each test example and illustrate the results in Figure 4. Red dots denote scores from the original PLMs, while blue dots are scores from DC-Match. We can observe that $P(y)$ and $Q(y)$ show much higher consistency (lower KL-Div. scores) when using the DC-Match strategy compared to the original PLMs, which again manifests the effectiveness of our devised divide-and-conquer training objective that narrows the gap between $P(y)$ and $Q(y)$.

5.5 Case Study

To intuitively understand how the DC-Match strategy works, we show several test cases of the QQP and MRPC datasets with predicted labels from different systems in Table 7. In order to analyze how the DC-Match enhanced PLMs make accurate predictions, we also show the solutions to the two sub-problems, namely $P(y_k|S_k^a, S_k^b)$ and $P(y_i|S_i^a, S_i^b)$, by directly introducing distant keyword labels as in Section 5.4. From the cases we observe that the final predictions of DC-Match are highly consistent with those of sub-problems. The model tends to output a low matching score as long as at least one of the sub-problems is inferred as mismatched. We also find that the original PLMs tend to make wrong predictions when two mismatched sentences share long common sub-sequences. For example, in the first case, the main difference between two sentences is the concept of ’cell’ and ’cell vacuoles’, but the remainder of the sequences is almost the same, which might confuse the model. By contrast, DC-Match is capable of identifying keywords from text sequences, and can make accurate judgements by dividing the matching problem into easier sub-problems.

6 Conclusion

In this work, we devise a divide-and-conquer training strategy DC-Match for text semantic matching. It breaks down the matching problem into two sub-problems: keyword matching and intent matching. The model learns to disentangle keywords from intents that require different levels of matching granularity. The proposed DC-Match is simple and effective, which can be easily combined with PLMs plus few additional parameters. We conduct experiments on three text matching datasets across different languages. Experimental results show that our approach can not only achieve stable performance improvement, but also shows robustness to semantically invariant text transformations.

[7] Here, we exploit the keyword labels in test sets only for analysis, and they do not influence model predictions.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61976056, 62076069).

References

Sumit Bhatia and Harit Vishwakarma. 2018. Know thy neighbors, and more! studying the role of context in entity recommendation. In *Proceedings of the 29th on Hypertext and Social Media*, pages 87–95.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE Transactions on Audio, Speech and Language Processing*.

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. 2020. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. In *NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First quora dataset release: Question pairs. *data. quora. com*.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6586–6593.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3403–3414.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. *arXiv preprint arXiv:2105.11136*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Varun Manjunatha, Nirat Saini, and Larry S Davis. 2019. Explicit bias discovery in visual question answering models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9554–9563. IEEE Computer Society.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10367–10378.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. 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.

Liang Pang, Yanyan Lan, and Xueqi Cheng. 2021. Match-ignition: Plugging pagerank into transformer for long-form text matching. *arXiv preprint arXiv:2101.06423*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020. 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)*, pages 2471–2486.

Yixuan Su, David Vandyke, Simon Baker, Yan Wang, and Nigel Collier. 2021. Keep the primary, rewrite the secondary: A two-stage approach for paraphrase generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 560–569.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *IJCAI*, pages 4411–4417.

Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018a. Co-stack residual affinity networks with multi-level attention refinement for matching text sequences. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4492–4502.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018b. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1565–1575.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International*

Conference on Neural Information Processing Systems, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Peilu Wang, Hao Jiang, Jingfang Xu, and Qi Zhang. 2019c. Knowledge graph construction and applications for web search and beyond. *Data Intelligence*, 1(4):333–349.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, et al. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349.

Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, Yanjie Fu, and Meng Wang. 2020. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 679–688.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709.

Kun Zhang, Enhong Chen, Qi Liu, Chuanren Liu, and Guangyi Lv. 2017. A context-enriched neural network method for recognizing lexical entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kun Zhang, Guangyi Lv, Linyuan Wang, Le Wu, Enhong Chen, Fangzhao Wu, and Xing Xie. 2019. Drrnet: Dynamic re-read network for sentence semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7442–7449.

Kun Zhang, Le Wu, Guangyi Lv, Meng Wang, Enhong Chen, and Shulan Ruan. 2021. Making the relation matters: Relation of relation learning network for sentence semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14411–14419.