DABERT: Dual Attention Enhanced BERT for Semantic Matching

Sirui Wang[1,*] , **Di Liang**[1,2,†] , **Jian Song**[†] , **Yuntao Li** [†], **Wei Wu**[†]
Tsinghua University, Beijing, China[*]
Meituan Inc.,Beijing, China[†]
{wangsirui,liangdi04,songjian20,liyuntao04,wuwei30}@meituan.com

Abstract

Transformer-based pre-trained language models such as BERT have achieved remarkable results in Semantic Sentence Matching. However, existing models still suffer from insufficient ability to capture subtle differences. Minor noise like word addition, deletion, and modification of sentences may cause flipped predictions. To alleviate this problem, we propose a novel **Dual Attention Enhanced BERT (DABERT)** to enhance the ability of BERT to capture fine-grained differences in sentence pairs. DABERT comprises (1) Dual Attention module, which measures soft word matches by introducing a new dual channel alignment mechanism to model affinity and difference attention. (2) Adaptive Fusion module, this module uses attention to learn the aggregation of difference and affinity features, and generates a vector describing the matching details of sentence pairs. We conduct extensive experiments on well-studied semantic matching and robustness test datasets, and the experimental results show the effectiveness of our proposed method.

1 Introduction

Semantic Sentence Matching (SSM) is a fundamental NLP task. The goal of SSM is to compare two sentences and identify their semantic relationship. In paraphrase identification, SSM is used to determine whether two sentences are paraphrase or not (Madnani et al., 2012). In natural language inference task, SSM is utilized to judge whether a hypothesis sentence can be inferred from a premise sentence (Bowman et al., 2015). In the answer sentence selection task, SSM is employed to assess the relevance between query-answer pairs and rank all candidate answers (Wang et al., 2020).

Across the rich history of semantic sentence matching research, there have been two main streams of studies for solving this problem. One is to utilize a sentence encoder to convert sentences into low-dimensional vectors in the latent space, and apply a parameterized function to learn the matching scores between them (Reimers and Gurevych, 2019; Wang et al., 2020). Another paradigm adopts attention mechanism to calculate scores between tokens from two sentences, and then the matching scores are aggregated to make a sentence-level decision (Chen et al., 2016; Tay et al., 2017). In recent years, pre-trained models, such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), have became much more popular and achieved outstanding performance in SSM. Recent work also attempts to enhance the performance of BERT by injecting knowledge into it, such as SemBERT (Zhang et al., 2020), UER-BERT (Xia et al., 2021), Syntax-BERT (Bai et al., 2021) and so on.

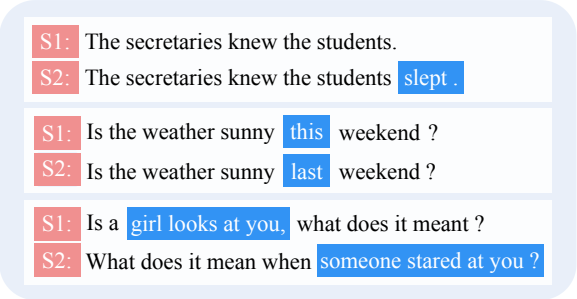

Figure 1: Example sentences with similar text but different semantics. S1 and S2 are sentence pair.

Although previous studies have provided some insights, those models do not perform well in distinguishing sentence pairs with high literal similarities but different semantics. Figure 1 demonstrates several cases suffering from this problem. Although the sentence pairs in this figure are semantically different, they are too similar in literal for those pre-trained language models to distinguish accurately. This could be caused by the self-attention architecture itself. Self-attention mechanism focuses on using the context of a word to understand the semantics of the word, while ignoring model-

[1]These authors contributed equally to this work.
[2]Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1645–1654
October 12–17, 2022.

ing the semantic difference between sentence pairs. De-attention (Tay et al., 2019) and Sparsegen (Martins and Astudillo, 2016) have proved that equipping with attention mechanism with more flexible structure, models can generate more powerful representations. In this paper, we also focus on enhancing the attention mechanism in transformer-based pre-trained models to better integrate difference information between sentence pairs. We hypothesize that paying more attention to the fine-grained semantic differences, explicitly modeling the difference and affinity vectors together will further improve the performance of pre-trained model. Therefore, two systemic questions arise naturally:

Q1: How to equip vanilla attention mechanism with the ability on modeling semantics of fine-grained differences between a sentence pair? Vanilla attention, or named affinity attention, less focuses on the fine-grained difference between sentence pairs, which may lead to error predictions for SSM tasks. An intuitive solution to this problem is to make subtraction between representation vectors to harvest their semantic differentiation. In this paper, we propose a dual attention module including a difference attention accompanied with the affinity attention. The difference attention uses subtraction-based cross-attention to aggregate word- and phrase- level interaction differences. Meanwhile, to fully utilize the difference information, we use dual-channel inject the difference information into the multi-head attention in the transformer to obtain semantic representations describing affinity and difference respectively.

Q2: How to fuse two types of semantic representations into a unified representation? A hard fusion of two signals by extra structure may break the representing ability of the pre-trained model. How to inject those information softly to pre-trained model remains a hard issue. In this paper, we propose an Adaptive Fusion module, which uses an additional attention to learn the difference and affinity features to generate vectors describing sentence matching details. It first inter-aligns the two signals through distinct attentions to capture semantic interactions, and then uses gated fusion to adaptively fuse the difference features. Those generated vectors are further scaled with another fuse-gate module to reduce the damage of the pre-trained model caused by the injection of difference information. The output final vectors can better describe the matching details of sentence pairs.

Our main contributions are three fold:

- We point out that explicitly modeling fine-grained difference semantics between sentence pairs can effectively benefit sentence semantic matching tasks, and we propose a novel dual attention enhanced mechanism based on BERT.
- Our proposed DABERT model uses a dual-channel attention to separately focus on the affinity and difference features in sentence pairs, and adopts a soft-integrated regulation mechanism to adaptively aggregate those two features. Thereby, the generated vectors can better describe the matching details of sentence pairs.
- To verify the effectiveness of DABERT, we conduct experiments on 10 semantic matching datasets and several data-noised dataset to test model's robustness. The results show that DABERT achieves an absolute improvement for over 2% compared with pure BERT and outperforms other BERT-based models with more advanced techniques and external data usage.

2 Approach

Our proposed DABERT is a modification of the original transformer structure, whose structure is shown in Figure 2. Two submodules are included in this new structure. (1) Dual Attention Module, which uses a dual channel mechanism in multi-head attention to match words between two sentences. Each channel uses a different attention head to calculate affinity and difference scores separately, and obtains two representations to measure affinity and difference information respectively. (2) Adaptive Fusion Module, which is used to fuse the representation obtained by dual attention. It first uses guide-attention to align the two signals. And then, multiple gate modules are used to fuse the two signals. Finally, a vector is output including more fine-grained matching details. In the following sections, we explain each component in detail.

2.1 Dual Attention Module

In this module, we use two distinct attention functions, namely affinity attention and difference attention, to compare the affinities and differences of vectors between two sentences. The input of the dual attention module is a triple of $K, Q, V \in R^{d_{seq} \times d_v}$, where d_v is the latent dimension and

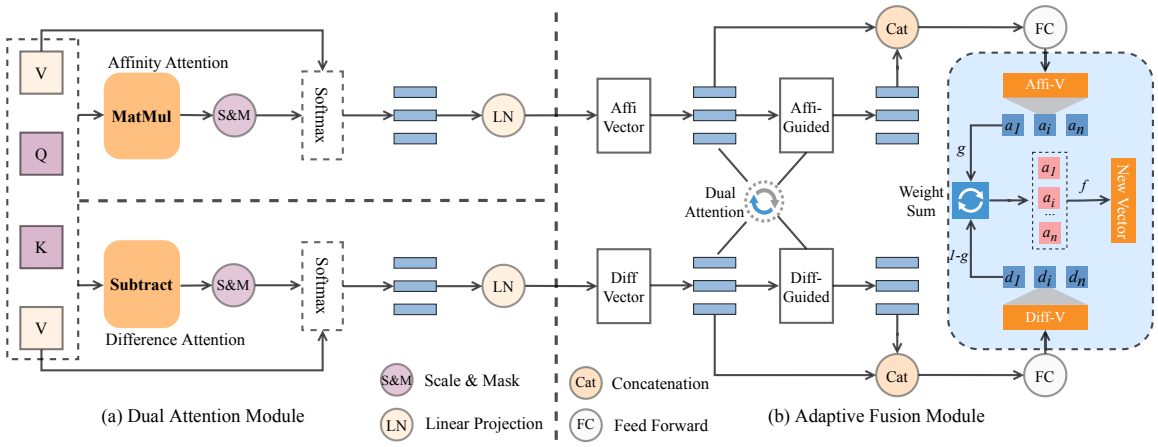

Figure 2: The overall architecture of Dual Attention Enhanced BERT (DABERT). The left side is the Dual attention module, and the right side is the Adaptive Fusion module.

d_{seq} is the utterance length. Dual attention module calculate the latent relationship between K, Q and V via two separate attention mechanism to measure their affinity and difference. As a result, two set of attention representations are generated by the dual attention module, which will be fused by the following adaptive fusion module.

2.1.1 Affinity Attention

The affinity attention module is the part of the dual attention module, which is the standard dot-product attention that operates following Transformer's default operation. The input of affinity attention module consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values. For the sake of simplicity, the formulations of BERT not be repeated here, please refer to (Devlin et al., 2018) for more details. We denote the output affinity vector as:

$$\mathbf{A} = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}) * \mathbf{V}, \tag{1}$$

where $\mathbf{A} = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_l\} \in R^{d_l \times d_v}$ denotes the vector describing affinity expressions generated by the Transformer original attention module.

2.1.2 Difference Attention

The second part of dual attention module is a difference attention module that capture and aggregate the difference information between sentence pairs. The difference attention module adopts a subtraction-based cross-attention mechanism, which allows model to pay attention to dissimilar parts between sentence pairs by element-wise subtraction as:

$$\mathbf{D} = softmax(\frac{\beta}{\sqrt{d_k}}) * \mathbf{V}, \tag{2}$$

$$\boldsymbol{\beta} = \|\mathbf{Q} - \mathbf{K}\| + \mathbf{M}, \tag{3}$$

$$\|\mathbf{Q} - \mathbf{K}\|_{ij} = \sum_{k=0}^{d_k} \mathbf{Q}_{ik} - \mathbf{K}_{jk}, \tag{4}$$

where $\|\mathbf{Q} - \mathbf{K}\| \in R^{d_l \times d_l}$ and d_l is the input sequence length. We use $\mathbf{D} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_l\} \in R^{d_l \times d_v}$ to denote the representation generated by the difference attention. The $\mathbf{M} \in R^{d_l \times d_l}$ is a masking operation. Both the affinity attention and the difference attention are utilized to fit the semantic relationship of sentence pairs, and obtain the representations with the same dimension from the perspective of affinity and difference respectively. This dual channel mechanism can obtain more detailed representations describing sentence matching.

2.2 Adaptive Fusion Module

After obtaining the affinity signals $\mathbf{A}$ and the difference signals $\mathbf{D}$, we introduce a novel adaptive fusion module to fuse these two signals instead of direct fusion (i.e., average embedding vector), since direct fusion may compromise the original representing ability of the pre-trained model. The fusion process includes three steps. First, it flexibly interacts and aligns these two signals via affinity-guided attention and difference-guided attention. Second, multiple gate modules are adopted to selectively extract interaction semantic information. Finally, to alleviate the damage of the pre-trained model by the difference signal, we utilize filter gates to adaptively filter out noisy information and finally

generate vectors that better describe the details of sentence matching.

Firstly, we update the difference vectors through affinity-guided attention. We use $\boldsymbol{a}_i$ and $\boldsymbol{d}_i$ to denote the i-th dimension of $\mathbf{A}$ and $\mathbf{D}$ respectively. We provide each affinity vector $\boldsymbol{a}_i$ to interact with the difference signals matrix $\mathbf{D}$ and obtain the new difference feature $\boldsymbol{d}_i^*$. Then, based on $\boldsymbol{d}_i^*$, we can in turn acquire the new Affinity feature $\boldsymbol{a}_i^*$ through difference-guided attention. The calculation process is as follows:

$$\begin{aligned}
\delta_i &= \tanh(\mathbf{W}_D\mathbf{D} \oplus (\mathbf{W}_{a_i}a_i + b_{a_i})),\\
\overline{d}_i &= \mathbf{D} * softmax(\mathbf{W}_{d_i}\delta_i + b_{d_i}),\\
\gamma_i &= \tanh(\mathbf{W}_A\mathbf{A} \oplus (\mathbf{W}_{\overline{d}_i}\overline{d}_i + b_{\overline{d}_i})),\\
\overline{a}_i &= \mathbf{A} * softmax(\mathbf{W}_{\overline{a}_i}\gamma_i + b_{\overline{a}*_i}),\\
d_i^* &= \tanh(\mathbf{W}_{d_i^*}([d_i;\overline{d}_i]) + b_{d_i^*})),\\
a_i^* &= \tanh(\mathbf{W}_{a_i^*}([a_i;\overline{a}_i]) + b_{a_i^*})),
\end{aligned} \tag{5}$$

where $\mathbf{W}_D, \mathbf{W}_A, \mathbf{W}_{a_i}, \mathbf{W}_{\overline{d}_i} \in R^{d_l * d_v}$; $\mathbf{W}_{d_i}$, $\mathbf{W}_{\overline{a}_i} \in R^{1*2d_l}$; $b_{d_i^*}$, b_{a_i}, b_{d_i}, $b_{a_i^*}$ are weights and bias of our model, and $\oplus$ denotes the concatenation of signal matrix and feature vector. Secondly, to adaptively capture and fuse useful information from Affinity and difference features, we introduce our gate fusion modules:

$$\begin{aligned}
\hat{d}_i &= \tanh(\mathbf{W}_{\hat{d}_i}d_i^* + b_{\hat{d}_i}),\\
\hat{a}_i &= \tanh(\mathbf{W}_{\hat{a}_i}a_i^* + b_{\hat{a}_i}),\\
g_i &= \sigma(\mathbf{W}_{g_i}(\hat{d}_i \oplus \hat{a}_i)),\\
v_i &= g_i\hat{a}_i + (1-g_i)\hat{d}_i,
\end{aligned} \tag{6}$$

where $\mathbf{W}_{\hat{d}_i}$, $\mathbf{W}_{\hat{a}_i} \in R^{d_h * d_v}$; $\mathbf{W}_{g_i} \in R^{1*2d_h}$; $b_{\hat{d}_i}$, $b_{\hat{a}_i}$ are parameters and d_h is the size of hidden layer. σ is the sigmoid activation function and g_i is the gate that determines the transmission of these two distinct representations. By the way, we get the fusion feature $\boldsymbol{v}_i$.

Eventually, considering the potential noise problem, we propose a filtering gate to selectively leverage the fusion feature. When $\boldsymbol{v}_i$ tends to be beneficial, the filtration gate will incorporate the fusion features and the original features. Otherwise, the fusion information will be filtered out:

$$\begin{aligned}
f_i &= \sigma(\mathbf{W}_{f_i,a_i}(a_i \oplus (\mathbf{W}_{v_i}v_i + b_{v_i}))),\\
l_i &= f_i * \tanh(\mathbf{W}_{l_i}v_i + b_{l_i}),
\end{aligned} \tag{7}$$

where $\mathbf{W}_{f_i,a_i} \in R^{1*2d_v}$; $\mathbf{W}_{v_i}$, $\mathbf{W}_{l_i} \in R^{d_v * d_h}$; b_{v_i}, b_{l_i} are trainable parameters and $\boldsymbol{l}_i$ is the final fused semantic feature and it will be propagated to the next computation flow.

3 Experimental Settings

3.1 Datasets

Semantic Matching. We conduct experiments on 10 sentence matching datasets to evaluate the effectiveness of our method. The GLUE (Wang et al., 2018) benchmark is a widely-used dataset in thie field, which includes tasks such as sentence pair classification, similarity and paraphrase detection, and natural language inference[1]. We conduct experiments on 6 sentence pair datasets (MRPC, QQP, STS-B, MNLI, RTE, and QNLI) from GLUE. We also conduct experiments on 4 other popular datasets (SNLI (Bowman et al., 2015), SICK (Marelli et al., 2014), TwitterURL (Lan et al., 2017) and Scitail (Khot et al., 2018)). The statistics of all 10 datasets are shown in Table 6.

Robustness Test. TextFlint (Gui et al., 2021) is a robustness evaluation platform for natural language processing models[2]. It includes more than 80 patterns to deform data, including inserting punctuation marks, changing numbers in text, replacing synonyms, modifying adverbs, deleting words, etc. It can effectively evaluate the robustness and generalization of models. In this paper, we leverage TextFlint to perform transformations on multiple datasets (Quora, SNLI, MNLI-m/mm), including task-specific transformations (SwapAnt, NumWord, AddSent) and general transformations (InsertAdv, Appendlrr, AddPunc, BackTrans, TwitterType, SwapNamedEnt, SwapSyn-WordNet). We conduct experiments on datasets with those types of transformations to verify the robustness of our model.

3.2 Baselines

To evaluate the effectiveness of our proposed DABERT in SSM, we mainly introduce BERT (Devlin et al., 2018), SemBERT (Zhang et al., 2020), SyntaxBERT (Liu et al., 2020), UERBERT (Xia et al., 2021) and multiple other PLMs (Radford et al., 2018; Devlin et al., 2018) for comparison. In addition, we also select several competitive models without pre-training as baselines, such as ESIM (Chen et al., 2016), Transformer (Vaswani et al., 2017) , etc (Hochreiter and Schmidhuber, 1997; Wang et al., 2017; Tay et al., 2017). In robustness experiments, we compare the performance of multiple pre-trained models (Sanh et al., 2019; Chen et al., 2016; Devlin et al., 2018; Lan et al., 2019)

[1]https://huggingface.co/datasets/glue
[2]https://www.textflint.io

Model	Pre-train	Sentence Similarity			Sentence Inference			Avg
		MRPC	QQP	SST-B	MNLI-m/mm	QNLI	RTE	
BiMPM†(Wang et al., 2017)	✗	79.6	85.0	-	72.3/72.1	81.4	56.4	-
CAFE†(Tay et al., 2017)	✗	82.4	88.0	-	78.7/77.9	81.5	56.8	-
ESIM†(Chen et al., 2016)	✗	80.3	88.2	-	-	80.5	-	-
Transformer†(Vaswani et al., 2017)	✗	81.7	84.4	73.6	72.3/71.4	80.3	58.0	74.53
BiLSTM+ELMo+Attn†(Devlin et al., 2018)	✓	84.6	86.7	73.3	76.4/76.1	79.8	56.8	76.24
OpenAI GPT†(Radford et al., 2018)	✓	82.3	70.2	80.0	82.1/81.4	87.4	56.0	77.06
UERBERT‡(Xia et al., 2021)	✓	88.3	90.5	85.1	84.2/83.5	90.6	67.1	84.19
SemBERT†(Zhang et al., 2020)	✓	88.2	90.2	87.3	84.4/84.0	90.9	69.3	84.90
BERT-base‡(Devlin et al., 2018)	✓	87.2	89.0	85.8	84.3/83.7	90.4	66.4	83.83
SyntaxBERT-base†(Bai et al., 2021)	✓	**89.2**	89.6	88.1	84.9/84.6	91.1	68.9	85.20
DABERT-base‡	✓	89.1	**91.3**	**88.2**	**84.9/84.7**	**91.4**	**69.5**	**85.58**
BERT-large‡(Devlin et al., 2018)	✓	89.3	89.3	86.5	86.8/85.9	92.7	70.1	85.80
SyntaxBERT-large†(Bai et al., 2021)	✓	**92.0**	89.5	88.5	86.7/86.6	92.8	74.7	87.26
DABERT-large‡	✓	91.4	**91.9**	**89.5**	**87.1/86.9**	**94.8**	**75.3**	**88.12**

Table 1: The performance comparison of DABERT with other methods. We report Accuracy $\times$ 100 on 6 GLUE datasets. Methods with † indicate the results from their papers, while methods with ‡ indicate our implementation.

Model	SNLI	Sci	SICK	Twi
ESIM†(Chen et al., 2016)	88.0	70.6	-	-
CAFE†(Tay et al., 2017)	88.5	83.3	72.3	-
CSRAN†(Tay et al., 2018)	88.7	86.7	-	84.0
BERT-base‡(Devlin et al., 2018)	90.7	91.8	87.2	84.8
UERBERT‡(Xia et al., 2021)	90.8	92.2	87.8	86.2
SemBERT†(Zhang et al., 2020)	90.9	92.5	87.9	86.8
SyntaxBERT-base†(Bai et al., 2021)	91.0	92.7	**88.7**	87.3
DABERT-base‡	**91.3**	**93.6**	88.6	**87.5**
BERT-large‡(Devlin et al., 2018)	91.0	94.4	91.1	91.5
SyntaxBERT-large†(Bai et al., 2021)	91.3	94.7	91.4	92.1
DABERT-large‡	**91.5**	**95.3**	**92.5**	**92.3**

Table 2: The performance comparison of DABERT with other methods on 4 popular datasets, including SNLI, Scitail(Sci), SICK and TwitterURL(Twi).

and SemBERT,UERBERT and Syntax-BERT on the robustness test datasets. For simplicity, the compared models are not described in detail here.

3.3 Implementation Details

DABERT is based on BERT-base and BERT-large. For distinct targets, our hyper-parameters are different. We use AdamW in the BERT and set the learning rate in $\{1e^{-}5, 2e^{-}5, 3e^{-}5, 8e^{-}6\}$. As for the learning rate decay, we use a warmup of 0.1 and L2 weight decay of 0.01. Furthermore, we set the epoch to 5 and the batch size is selected in {16, 32, 64}. We also set dropout at 0.1-0.3. To prevent gradient explosion, we set gradient clipping in {7.5, 10.0, 15.0}. All the experiments are conducted by Tesla V100 and PyTorch platform. In addition, to ensure that the experimental results are statistically significant, we conduct each experiment five times and report the average results.

4 Results and Analysis

4.1 Model Performance

In our experiments, we implement DABERT in the initial transformer layer of BERT.

First, we fine-tune our model on 6 GLUE datasets. Table 1 shows the performance of DABERT and other competitive models. It can be seen that using only non-pretrained models performs obviously worse than PLMs due to their strong context awareness and data fitting capabilities. When the backbone model is BERT-base or BERT-large, the average accuracy of DABERT respectively improves by 1.7% and 2.3% than vanilla BERT. Such great improvement demonstrates the benefit of fusion difference attention for mining semantics and proves that our framework can help BERT perform much better in SSM.

Moreover, compared with some previous works such as SemBERT, UERBERT and SyntaxBERT, DABERT achieves the best performance without injecting external knowledge. Specifically, our model outperforms SyntaxBERT, the best performing model in previous work leveraging external knowledge, with an average relative improvement of 0.86% based on BERT-large. On the QQP dataset, the accuracy of DABERT is significantly improved by 2.4% over SyntaxBERT. There are two main reasons for such results. On the one hand, we use dual-channel attention to enhance the ability of DABERT to capture difference features. This enables DABERT to obtain more fine-grained interaction matching features. On the other hand, for

Model	Quora					SNLI				
	SA	NW	IA	Al	BT	AS	SA	TT	SN	SW
ESIM†(Chen et al., 2016)	-	-	-	-	-	64.00	84.22	78.32	53.76	65.38
BERT‡(Devlin et al., 2018)	48.58	56.96	86.32	**85.48**	83.42	79.66	94.84	83.56	50.45	76.42
ALBERT‡(Lan et al., 2019)	51.08	55.24	81.87	78.94	82.37	45.17	96.37	81.62	57.66	74.93
UERBERT‡(Xia et al., 2021)	48.57	54.86	84.72	80.88	82.71	73.24	94.78	85.36	57.54	80.81
SemBERT‡(Zhang et al., 2020)	50.92	53.15	85.19	82.04	82.40	76.81	95.31	84.60	56.28	77.86
SyntaxBERT‡(Bai et al., 2021)	49.30	56.37	86.43	84.62	84.19	78.63	95.02	**86.91**	58.26	76.90
DABERT‡	**60.43**	**62.76**	**87.50**	85.48	**87.49**	**81.06**	**96.85**	85.14	**60.58**	**80.92**

Method	MNLI-m/mm					
	AS	SA	AP	TT	SN	SW
BERT‡(Devlin et al., 2018)	55.32/55.25	52.76/55.69	82.30/82.31	77.08/77.22	51.97/51.84	76.41/77.05
ALBERT‡(Lan et al., 2019)	53.09/53.58	50.25/50.20	**83.98/83.68**	**77.98**/78.03	56.43/50.03	76.63/77.43
UERBERT‡(Xia et al., 2021)	54.99/54.84	52.29/53.80	79.80/79.18	75.46/74.93	55.21/55.96	**82.23**/82.74
SemBERT‡(Zhang et al., 2020)	55.38/55.12	54.07/54.62	78.70/78.16	73.90/73.47	53.43/53.76	78.09/78.93
SyntaxBERT‡(Bai et al., 2021)	54.92/54.63	53.54/54.73	77.01/76.71	70.38/70.13	57.11/51.95	78.57/79.31
DABERT‡	**60.14/59.25**	**60.89/61.37**	83.23/83.19	77.94/**78.10**	**60.12/59.83**	82.15/**82.97**

Table 3: The robustness experiment results of DABERT and other models. The data transformation methods we utilized mainly include SwapAnt (SA), NumWord (NW), AddSent (AS), InsertAdv (IA), Appendlrr (Al), AddPunc (AP), BackTrans (BT), TwitterType (TT), SwapNamedEnt (SN), SwapSyn-WordNet (SW).

the potential noise problem introduced by external structures, our adaptive fusion module can selectively filter out inappropriate information to suppress the propagation of noise, and previous work does not seem to pay enough attention to this problem. However, we still notice that SyntaxBERT achieves slightly better accuracy on a few datasets. We argue that this is a result of the intrinsic correlation of syntactic and dependent knowledge.

Second, to verify the general performance of our method, we also conduct experiments on other popular datasets. The results are shown in Table 2. DABERT still outperforms vanilla BERT and other models on almost all datasets. It is worth noting that DABERT performs worse than SyntaxBERT on SICK. This may be because the data volume of SICK is relatively small, and SyntaxBERT uses syntactic prior knowledge, which makes SyntaxBERT more advantageous on small datasets. but DABERT still shows a very competitive performance on SICK, which also shows from the side that our method can enhance the difference capture ability of BERT and make up for the lack of generalization ability with fewer parameters.

Overall, our method has competitive performance in judging semantic similarity compared to previous work. Extensive performance improvements also validate our point, soft ensemble difference information based on BERT's powerful contextual representation capability is useful for sentence matching tasks.

4.2 Robustness Test Performance

In order to examine the performance of DABERT and competitive models in their ability to capture subtle differences in sentence pairs. We perform robustness tests on three extensively studied datasets.

Table 3 lists the accuracy of DABERT and six baseline models on the three datasets. We can observe that SwapAnt leads to a drop in maximum performance, and our model outperforms the best model SemBert nearly 10% on SwapAnt(QQP), which indicates that DABERT can better handle semantic contradictions caused by antonyms than baseline models. And the model performance drops to 56.96% on NumWord transformation, while DABERT outperforms BERT by nearly 6% because it requires the model to capture subtle numerical differences for correct linguistic inference. In SwapSyn transformation, UERBERT significantly outperforms other baseline models because it explicitly uses the synonym similarity matrix to calibrate the attention distribution, while our model can still achieve comparable performance to UERBERT without adding external knowledge. On TwitterType and AddPunc, the performance of SyntaxBERT by injecting syntax trees degrades significantly, probably because converting text to twitter type or adding punctuation breaks the normal syntactic structure of sentences. And DABERT still achieves the competitive performance in these two transformations. In other scenarios, DABERT also achieve better performance due to capturing subtle

Case	ESIM	BERT	SyntaxBERT	DABERT
S1:How done you solve this aptitude question? S2:How does I solve aptitude questions on cube?	label:1	label:0	label:0	similarity:10.87% label:0
S1:How can I tell if this girl loves me? S2:How can I tell if this boy loves me?	label:1	label:1	label:1	similarity:12.06% label:0
S1:How many 12 digits number have the sum of 4? S2:How many 42 digits number have the sum of 4?	label:1	label:1	label:1	similarity:18.63% label:0

Table 4: The example sentence pairs of our cases. Red and Blue are difference phrases in sentence pair.

Model	Quora		QNLI	
	Dev	Test	Dev	Test
DABERT	**92.1**	**91.3**	**92.9**	**91.4**
w/o Affi-attention	90.1	89.5	91.8	90.7
w/o Diff-attention	90.6	89.8	92.0	90.8
w/o Guide-attention	91.3	90.4	92.1	91.0
w/o Gate fusion	91.7	90.6	92.5	91.1
w/o Gate filter	91.8	90.9	92.6	91.2
w/o Regulation	89.9	89.4	91.5	90.7

Table 5: Results of component ablation experiment.

differences in sentence pairs. Meanwhile, ESIM has the worst performance, the results reflect that the pre-training mechanism benefits from rich external resources and provides better generalization ability than de novo trained models. And the improved pre-trained model SyntaxBERT performs better than the original BERT model, which reflects that sufficient pre-trained corpus and suitable external knowledge fusion strategies can help improve the generalization performance of the model.

4.3 Ablation Study

To evaluate the contribution of each component in our method, we conduct ablation experiments on the QQP and QNLI datasets based on BERT. The experimental results are shown in the table 5.

Above all, the dual attention module consists of two core components that use a two-channel mechanism to model affinity and difference attention. First, after removing affinity attention, the performance of the model on the two datasets drops by 1.8% and 0.7%. Affinity attention can capture the dynamic alignment relationship between word pairs, which is crucial for SSM tasks. Next,after removing difference attention from the model, the performance on the two datasets dropped by 1.5% and 0.6%, respectively. The difference information can further describe the interaction between words, and can provide more fine-grained comparison information for the pre-trained model, so that the model can obtain a better representation. The above experiments show that the performance drops sharply after the submodule is removed, which demonstrates the effectiveness of the internal components of the dual attention module.

Next, in the adaptive fusion module, we also conducted several experiments to verify the effect of the fusion of affinity and difference vectors. On the QQP dataset, we first remove the guide attention module, and the performance drops to 90.4%. Since guide attention can capture the interaction between two signals, this interaction information is crucial for fusing two different information. Second, after removing the fusion gate, we only integrate two signals by simple averaging. The accuracy dropped to 91.4%, indicating that dynamically merging the affinity and difference vectors according to different weights can improve the performance of the model. Then, when the filter gate is removed, the accuracy drops by 0.4%, indicating that the ability of the model to suppress noise is weakened without the filter gate. Finally, we also replaced the overall aggregation and Regulation module with simple average, and the performance dropped sharply to 89.4%. While difference information is crucial for judging sentence-pair relations, hard-integrating the difference information into the PLMs will destroy its Pre-existing knowledge, and soft aggregation and governance can make better use of difference signals.

Overall, due to the effective combination of each component, DABERT can adaptively fuse difference features into pretrained models and leverage its powerful contextual representation to better inference about semantics.

4.4 Case Study

To visualize how DABERT works, we use three cases from the table 4 for qualitative analysis. In the first case, the non-pretrained language model ESIM is difficulty capturing the semantic conflicts caused by the difference words. Therefore, ESIM

Datasets	#Train	#Dev	#Test	#Class
MRPC	3669	409	1380	2
QQP	363871	1501	390965	2
MNLI-m/mm	392703	9816/9833	9797/9848	3
QNLI	104744	40432	5464	2
RTE	2491	5462	3001	2
SST-B	5749	1500	1379	2
SNLI	549367	9842	9824	3
SICK	4439	495	4906	3
Scitail	23596	1304	2126	2
TwitterURL	42200	3000	9324	2

Table 6: The statistics of all 10 datasets.

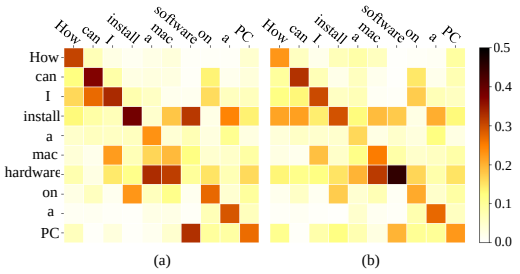

Figure 3: Distribution of BERT (a) and our method (b).

gives wrong prediction results in case 1. BERT can identify the semantic difference in case 1 with the help of context representation . But in case 3, BERT cannot capture the difference between the numbers "12" and "24" and give wrong prediction. SyntaxBERT enhances text understanding by introducing syntactic trees. Since case 2 and case 3 have the same syntactic structure, SyntaxBERT also gives wrong predictions. Our model made correct predictions in all of the above cases. Because DABERT explicitly focuses on different parts of sentence pairs through difference attention and adaptively aggregates affinity and difference information in the adaptive fusion module, it can identify semantic differences caused by subtle differences within sentence pairs.

Attention Distribution. To verify the fusion effect of subtraction-based attention on the difference information, we display the weights distribution of BERT and DABERT in Figure 3 for comparison. It can be seen that the attention distribution after dual attention becomes more reasonable, especially the attention weight between "hardware" and "software" increases significantly. This reveals that DABERT pays more attention to different parts of sentence pairs rather than the same words.

5 Related Work

Semantic Sentence Matching is a fundamental task in NLP. Thanks to the appearance of large-scale annotated datasets (Bowman et al., 2015; Williams et al., 2017), neural network models have made great progress in SSM (Qiu and Huang, 2015; Wan et al., 2016), mainly fell into two categories. The first (Conneau et al., 2017; Choi et al., 2018) focuses on encoding sentences into corresponding vectors without cross-interaction and applies a classifier to obtain similarity. The second (Wang et al., 2017; Chen et al., 2016; Liang et al., 2019a) utilizes cross-features as an attention module to express the word- or phrase-level alignments of two texts, and aggregates it into prediction layer to acquire similarity. Recently, the pre-training paradigm has achieved great results in SSM. Some work attempt to introduce other methods to enhance pre-trained models. For example, SemBERT (Zhang et al., 2020) explicitly absorbs contextual semantics over a BERT backbone. AMAN (Liang et al., 2019b) uses answers knowledge to enhance language representation. UER-BERT (Xia et al., 2021) injects synonym knowledge to enhance BERT. SyntaxBERT (Bai et al., 2021) also integrates the syntax tree into transformer models.

Robustness Although neural network models have achieved human-like or even superior results in multiple tasks, they still face the insufficient robustness problem in real application scenarios (Gui et al., 2021). Tiny literal changes may cause misjudgments. Therefore, recent work starts to focus on robustness research from multiple perspectives. TextFlint (Gui et al., 2021) incorporates multiple transformations to provide comprehensive robustness analysis. Li et al. (2021) provide an overall benchmark for current work on adversarial attacks. And Liu et al. (2021) propose a more comprehensive evaluation system and add more detailed output analysis indicators.

6 Conclusions

In this paper, we propose a novel Dual Attention Enhanced BERT (DABERT), which can efficiently aggregate the difference information in sentence pairs and soft-integrate it into a pretrained model. Based on BERT's powerful contextual representation capability, DABERT enables the model to learn more fine-grained comparative information and enhances the sensitivity of PLMs to subtle differences. Experimental results on 10 public datasets and robustness dataset show that our method can achieve better performance than several strong baselines. Since DABERT is an end-to-end training component, it is expected to be applied to other large-scale pre-trained models in the future.

References

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-bert: Improving pre-trained transformers with syntax trees. *arXiv preprint arXiv:2103.04350*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tao Gui, Xiao Wang, Qi Zhang, Qin Liu, Yicheng Zou, Xin Zhou, Rui Zheng, Chong Zhang, Qinzhuo Wu, Jiacheng Ye, et al. 2021. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. *arXiv preprint arXiv:2103.11441*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*.

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. *arXiv preprint arXiv:1909.11942*.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. *arXiv preprint arXiv:2108.12777*.

Di Liang, Fubao Zhang, Qi Zhang, and Xuan-Jing Huang. 2019a. Asynchronous deep interaction network for natural language inference. 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)*, pages 2692–2700.

Di Liang, Fubao Zhang, Weidong Zhang, Qi Zhang, Jinlan Fu, Minlong Peng, Tao Gui, and Xuanjing Huang. 2019b. Adaptive multi-attention network incorporating answer information for duplicate question detection. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 95–104.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, Zi-Yi Dou, and Graham Neubig. 2021. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

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. *arXiv preprint arXiv:1907.11692*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190, Montréal, Canada. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR.

Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *Twenty-Fourth international joint conference on artificial intelligence*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Yi Tay, Anh Tuan Luu, Aston Zhang, Shuohang Wang, and Siu Cheung Hui. 2019. Compositional de-attention networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*, 78:154.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Co-stack residual affinity networks with multi-level attention refinement for matching text sequences. *arXiv preprint arXiv:1810.02938*.

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.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Shuohang Wang, Yunshi Lan, Yi Tay, Jing Jiang, and Jingjing Liu. 2020. Multi-level head-wise match and aggregation in transformer for textual sequence matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9209–9216.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2021. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, pages 2466–2475.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.