Leveraging Multiway Attention to Enhance Language Model for Semantic Matching

Anonymous NAACL 2025 submission

Abstract

Semantic matching is a fundamental task in Natural Language Processing (NLP), which is widely used in information retrieval, recommendation, and other applications. Transformer-based pre-trained language models have achieved remarkable improvements in semantic matching. However, the transformer uses only one attention mechanism, which might not be optimal for semantic matching that relies on the modeling of complex relationships. In this paper, we propose the Multiway Attention and Adaptive Fusion mechanism to enhance the ability of language models to capture the relationships between sentence pairs from diverse aspects by exploiting four complementary attention mechanisms. Building based upon the transformer architecture, the method adopts diverse types of attention functions to capture manifold types of interactive information and effectively fuses them with a well-designed self-interactive augmentation layer and a normalized aggregation layer. Specifically, the multiway attention mechanism includes three key modules, 1) a multiway attention module, 2) a self-interactive augmentation module, and 3) a normalized aggregation module. We apply the proposed mechanism to language models and conduct extensive experiments. Results show that the proposed model achieves consistent improvement on 10 well-studied semantic matching datasets.

1 Introduction

Semantic Sentence Matching (SSM) is a fundamental task in Natural Language Processing. SSM aims to compare two sentences and identify their semantic relationship. For example, in paraphrase identification, SSM is used to determine whether two sentences are paraphrased or not (Madnani et al., 2012). In natural language inference tasks, 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).

Figure 1: Examples of other PLMs like BERT and GPT 4o mini can not distinguish the semantics of similar texts well.

In recent years, with the development of pre-trained language models(PLMs), PLMs with attention are regarded as the core structure, such as BERT(Devlin et al., 2018), RoBERTa(Liu et al., 2019b). Recent work shows that using external knowledge to enhance the attention mechanism can further improve the performance of the model. For example, SyntaxBERT(Bai et al., 2021) proposes to exploit the text sentence structure and enhance the model by adding syntactic information to the attention. By introducing synonym information to enhance the attention mechanism in the pre-trained language model, UERBERT(Xia et al., 2021) achieves significant performance improvements. These works show the importance of incorporating some inductive bias into the attention mechanism for text sequence learning.

Most of the current pre-trained models are based on transformer architecture. The calculation of the attention score in the transformer is merely based on the dot-product to model the relationship between sentence pairs, which may not be optimal for the transformer-based pre-trained model. These models do not perform very well in distinguishing sentence pairs with high literal similarities. Figure 1 demonstrates a few cases suffering from this problem. Although the sentence pairs in this figure are semantically different, they are too similar in literally for those language models like BERT and GPT 4o to distinguish accurately. which reserves space for improving the transformer-based pre-trained model. At the same time, De-attention (Tay et al., 2019) and MwAN (Tan et al., 2018) work in the non-pre-trained model have verified the effectiveness of a flexible attention mechanism. Therefore, we design a more flexible attention model and incorporate it into the pre-trained model to achieve improved performance.

To this end, in this paper, we propose a new method, named Multiway Attention and Adaptive Fusion Mechanism. BERT is a representative model of bidirectional language models, so we apply our proposed mechanism in BERT, as the attention in the first layer transformer of BERT is broad and uninformed(Xia et al., 2021), meanwhile, so as to avoid adding unnecessary parameters, The proposed model reforms the multi-head attention module of the first layer transformer of BERT by incorporating four complementary attention mechanisms, interactively augment and adaptively integrate four kinds of attention information for sentence matching. due to the fact that the attention information is captured from four perspectives or views, we call our attention mechanism: *Multiway Attention and Adaptive Fusion*. Specifically, it includes three modules:

1) Multiway Attention module. We analyze that the four kinds of attention are complementary clues for sentence matching, which can capture different levels of information in the text sequence. We propose a multiway attention model by considering four different attention mechanisms, including dot-product attention, additive attention, minus attention, and bilinear attention. The additive attention module for natural language reasoning, and the bilinear attention module for reading comprehension. In addition, we also use the point-to-point difference of vectors, minus attention, to calculate the relationship between words. Compared with the single attention-based model, our framework can model the relationship between sentences from different perspective through multiway attention, so as to obtain more fine-grained matching information.

2) Self-interactive Augmentation module. We observe that The representations obtained through different attention mechanisms do not interact well with input information, We propose to apply the self-interactive augmentation module to interact with the matching information together with each word in the sentence in each attention function. So in order to augment the interaction with input information and thus obtain better semantic representations.

3) Normalized Aggregation module. We find that the simple aggregation with fixed or average importance weights may destroy the learned knowledge of the pre-trained language model. We propose to adaptively aggregate the representations obtained by the self-interactive augmentation module, normalized aggregation combines the matching information of all attention functions. We apply the normalized aggregation module to aggregate the four representations adaptively.

In order to verify the effectiveness of our proposed model, we conducted intensive experiments on 10 datasets, including GLUE datasets such as QQP, MRPC, and SNLI, and fully studied datasets such as Sci and Twi. The results show that compared with BERT-base, our model achieves an absolute improvement of more than 2.2% avg, and is superior to other BERT-based models and large language models(LLMs) in more advanced technology and external data use.

2 Related Work

Text semantic matching plays an important role in many applications, such as information retrieval (IR) and natural language Inference (NLI). And the transformer model(Vaswani et al., 2017) to enhance the matching performance based on the attention mechanism. In addition, some models introduce some task-related information for specific tasks to help the model reasoning, such as BIMPM(Wang et al., 2017) and ADIN(Liang et al., 2019a) for adding two-way semantic matching, ESIM(Chen et al., 2016) for local reasoning and reasoning combination layer, DPM-Net(Xue et al., 2023) and AMAN(Liang et al., 2019b). for a high-level un-

derstanding of sentences, Which shows the effectiveness of task-oriented approvals and model designing strategies based on information interaction.

Recently, large-scale pre-trained language models (PLMs) 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 attention and are pre-trained with well-designed self-supervised learning objectives. Outstanding examples include Embedding from Language Models(ELMo) (Peters et al., 2018), Generative Pre-trained Transformers (GPT) (Radford et al., 2018), Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), and Generalized Auto-regressive Pre-training (XLNet) (Yang et al., 2019). Providing fine-grained contextual word embedding, these pre-trained models can be either easily applied to downstream tasks as encoders or directly fine-tuned for downstream tasks. As the most prominent model in recent years, BERT and many of its variants, including AlBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019b), ERNIE (Zhang et al., 2019), K-BERT (Liu et al., 2020), DeBERTa(He et al., 2020), DABERT(Wang et al., 2022), DC-Match(Zou et al., 2022), DAFA(Song et al., 2022) and QC-MHM(Xue et al., 2024) have achieved superior results in many NLP tasks.

Although the pre-trained model shows a strong representation ability in sentence encoding, there are still some improvements for multi-head attention, which is used to improve the coding ability of the pre-trained model and improve the performance effect on downstream tasks, Such as 1) SyntaxBERT(Bai et al., 2021) improves the model's understanding of text sentence structure by adding syntactic information to attention, 2) UERBERT(Xia et al., 2021) enhances the attention mechanism in the pre-trained model by introducing synonym information, and 3) SemBERT(Zhang et al., 2020) improves the effect of text representation by integrating semantic role tagging and multi-label semantics into attention. In this work, we propose a scheme, which is used to provide different attention modes to capture the relationship between different components in a sentence, interactively augment and aggregate the four different attention modes **(Multiway Attention)** is used to improve the encoding ability of multi-head attention for text, and significantly improve the matching of short text, which can be easily combined with PLMs to stack additional improvements for text semantic matching.

3 Approach

We show the Multiway Attention and Adaptive Fusion mechanism in Figure 2. We take BERT as our base model, According to the findings in (Xia et al., 2021), the attention in the first layer transformer of BERT is broad and uninformed, so in order to improve the performance of the BERT as much as possible without adding extra parameters, We decided to apply our multiway attention mechanism only in the first transformer layer of BERT. Regarding this point, we also compared the results of changes in attention in each layer and changes in attention in all layers in the experiments. It also indirectly verified that adding only changes to attention in the first layer is optimal, please refer to the A.2 for more details. Which consists of four parts under the augmentation-aggregation framework. Specifically, For every word embedding from **q** and **k**, we can obtain four matching scores using four different attention functions. Next, we augment the matching information along with words in **q**. We match two vectors inside each attention function interactively and then combine the matching information from all functions. The Multi-Layer Perception(MLP) is applied to fuse the matching information both in the self-interactive augmentation and normalized aggregation. Finally, that is, we obtain an aggregated multi-dimensional attention result after multiway attention, self-interactive augmentation, and normalized aggregation. which is used to replace the self-attention part in the first layer transformer of BERT.

3.1 Multiway Attention

In the multiway attention module, we use four different attention functions to model the semantic relationship between sentence pairs from different perspectives. Note that this is in stark contrast to the single Attention used by default in the transformer. The four attentions are dot attention, additive attention, minus attention, and bilinear attention, respectively. The input of the multiway attention module is a triple of Q, K, $V \in R^{d_{seq}} \times d_v$, where d_v is the latent dimension, d_{seq} is the length of the utterance. We use q_i, k_i and v_i to denote the i-th dimension of Q, K, and V respectively. Four independent attention mechanisms compute poten-

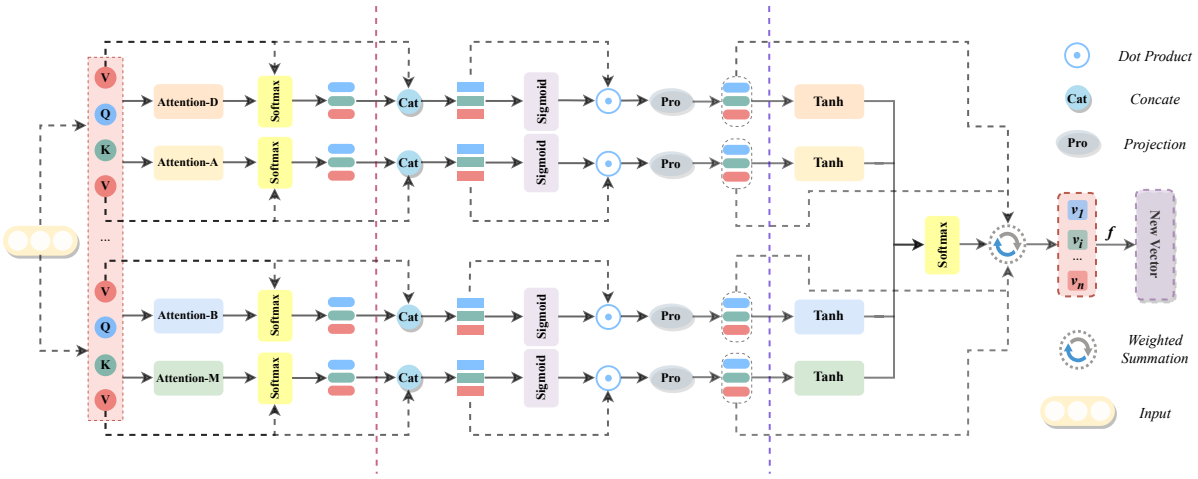

Figure 2: The detailed structures of the Multiway Attention module, Self-interactive Augmentation module, and Normalized Aggregation module.

tial relationships between Q, K, and V to measure their semantic interaction alignment.

3.1.1 Dot Attention

Dot attention is the first part of the multiway attention module, which can directly compute correlations using matrix operations, and the computed scores are correlation weights. It is also the most commonly used attention mechanism in semantic correlation modeling. And it follows the standard dot-product attention that the transformer operates by default. The input of the dot 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, 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 vector as:

$$s_j^t = \mathbf{q}_j \odot \mathbf{k}_t \tag{1a}$$

$$a_i^t = \frac{exp(s_i^t)}{\sum_{j=1}^{N} exp(s_j^t)} \tag{1b}$$

$$\mathbf{q}_t^d = \sum_{i=1}^{N} a_i^t \mathbf{v}_i \tag{1c}$$

where $\mathbf{q}_t^d \in R^{1\times d_v}$ is the output of the t-th position obtained after the dot attention calculation and $\odot$ is element-wise dot product.

3.1.2 Additive Attention

The second part of multiway attention is the additive attention module, which is more inclined to capture aligned representations between sentence pairs from a global perspective since it applies a concatenated alignment of two vectors. Specifically, its input is the concatenation of two vectors, the output after the activation function represents the correlation between the vectors, and finally, the softmax function is applied to get the weights of the values. We denote the output vector as:

$$s_j^t = \tanh(\mathbf{W}_a([\mathbf{q}_j; \mathbf{k}_t]) \tag{2a}$$

$$a_i^t = \frac{exp(s_i^t)}{\sum_{j=1}^{N} exp(s_j^t)} \tag{2b}$$

$$\mathbf{q}_t^a = \sum_{i=1}^{N} a_i^t \mathbf{v}_i \tag{2c}$$

where $\mathbf{q}_t^a \in R^{1\times d_v}$ is the output of the t-th position obtained after the additive attention calculation, and $\mathbf{W}_a \in R^{1\times 2d_v}$ are weights of our model.

3.1.3 Minus Attention

The third part of multiway attention is the minus attention module which captures and aggregates the different information between sentence pairs. The difference attention module adopts a subtraction-based cross-attention mechanism, which allows the model to pay attention to dissimilar parts between sentence pairs by element-wise subtraction as:

$$s_j^t = \tanh(\mathbf{W}_m(\mathbf{q}_j - \mathbf{k}_t)) \tag{3a}$$

$$a_i^t = \frac{exp(s_i^t)}{\sum_{j=1}^{N} exp(s_j^t)} \tag{3b}$$

$$\mathbf{q}_t^m = \sum_{i=1}^{N} a_i^t \mathbf{v}_i \tag{3c}$$

where $\mathbf{q}_t^m \in R^{1\times d_v}$ is the output of the t-th position obtained after the minus attention calculation, and $\mathbf{W}_m \in R^{1\times d_v}$ are weights of our model.

3.1.4 Bilinear Attention

The last part of multiway attention is a bilinear attention module, which can learn a bilinear attention distribution of two vectors to seamlessly utilize the given sentence pair information. It models the bilinear interaction between two sets of input channels, facilitating the extraction of a joint representation of each pair of channels. Its calculation formula is as follows:

$$s_j^t = \mathbf{q}_j^T \mathbf{W}_b \mathbf{k}_t \tag{4a}$$

$$a_i^t = \frac{exp(s_i^t)}{\sum_{j=1}^{N} exp(s_j^t)} \tag{4b}$$

$$\mathbf{q}_t^b = \sum_{i=1}^{N} a_i^t \mathbf{v}_i \tag{4c}$$

where $\mathbf{q}_t^b \in R^{1\times d_v}$ is the output of the t-th position obtained after the bilinear attention calculation and $\mathbf{W}_b \in R^{d_v\times d_v}$ are weights of our model.

3.2 Self-interactive Augmentation

Self-interactive augmentation is to fuse each word in the query vector in each attention function. For each position t, we splice the word representation h_t^k of v_t with its representation q_t^c of the corresponding attention, where v_t denotes the t-th dimension of K, c = (a, b, d, m). which will better augment each word representation with single attention to capture the input information. then using a gating structure to select the importance after splicing, and then applying an MLP to fuse the representation of each position more fully after the above operations. After that, the output after self-interactive augmentation is obtained. As shown below, This is an example of our self-interactive augmentation of additive attention:

$$\mathbf{x}_t^a = \left[q_t^a, h_t^k\right] \tag{5a}$$

$$g_i = sigmoid\,(\mathbf{W}_g \mathbf{x}_t^a) \tag{5b}$$

$$\mathbf{x}_t^{a*} = g_i \odot \mathbf{x}_t^a \tag{5c}$$

$$\mathbf{h}_t^a = \tanh(\mathbf{W}_d \mathbf{x}_t^{a*} + b_a) \tag{5d}$$

For bilinear, dot, and minus attention, we will also get h_t^b, h_t^d, and h_t^m, respectively. Where $\mathbf{W}_g \in R^{1\times 2d_v}$, $\mathbf{W}_d \in R^{d_v\times 2d_v}$, b_a are weights and bias of our model.

3.3 Normalized Aggregation

Normalized aggregation is to fuse all the attention functions. We use a parameter z^c as an input to adaptively fuse four different attention mechanisms.

$$s_j = \tanh(\mathbf{W}_1 h_t^j + \mathbf{W}_2 \mathbf{z}^c)(j = a, b, d, m) \tag{6a}$$

$$a_i = \frac{exp(s_i)}{\sum_{j=(a,b,d,m)} exp(s_j)} \tag{6b}$$

$$\mathbf{x}_t = \sum_{i=(a,b,d,m)} a_i \mathbf{h}_t^i \tag{6c}$$

Then, we input this X_t into an MLP neural network to fuse the information in different attention functions, and we will obtain different h_t^o for different positions in P.

$$\mathbf{h}_t^o = \tanh(\mathbf{W}_t \mathbf{x}_t + b_t) \tag{7}$$

Where $\mathbf{W}_t \in R^{d_v\times d_v}$, b_t are weights and bias of our model, respectively.

After aggregating the multiway attention matching information, we will obtain the fused representation of the vectors at different positions for t from 1 to N.

$$A^o = (h_1^o, h_2^o, ..., h_N^o) \tag{8}$$

A^o is the final fused semantic feature and it will be propagated to the next computation flow.

4 Experiment

The datasets, baselines, and all details of our experiments are shown in Appendix A.4.

Table 1: The performance comparison of our method 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.

Method	**Pre-trained**	**MRPC**	**QQP**	**MNLI-m/mm**	**QNLI**	**RTE**	**STS-B**	**Avg**
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	72.3/71.4	80.3	58.0	73.6	74.53
BiLSTM+ELMo+Attn†(Peters et al., 2018)	✓	84.6	86.7	76.4/76.1	79.8	56.8	73.3	76.24
OpenAI GPT†(Radford et al., 2018)	✓	82.3	70.2	82.1/81.4	87.4	56.0	80.0	77.06
UERBERT‡(Xia et al., 2021)	✓	88.3	90.5	84.2/83.5	90.6	67.1	85.1	84.19
SemBERT†(Zhang et al., 2020)	✓	88.2	90.3	84.4/84.0	90.9	69.3	87.3	84.90
BERT-base‡(Devlin et al., 2018)	✓	87.2	89.0	84.3/83.7	90.4	66.4	85.8	83.83
RoBERTa‡(Liu et al., 2019b)	✓	87.9	89.2	84.7/84.1	90.7	67.2	86.7	84.43
SyntaxBERT-base†(Bai et al., 2021)	✓	**89.2**	89.6	84.9/84.6	91.1	68.9	88.1	85.20
ours-base‡	✓	89.1	**92.0**	**84.9/85.3**	**92.1**	**69.8**	**88.9**	**86.13**
BERT-large‡(Devlin et al., 2018)	✓	89.3	89.3	86.8/85.9	92.7	70.1	86.5	85.80
RoBERTa-large‡(Devlin et al., 2018)	✓	90.4	89.4	86.8/86.1	92.7	72.3	87.5	86.32
SyntaxBERT-large†(Bai et al., 2021)	✓	**92.0**	89.5	86.7/86.6	92.8	74.7	88.5	87.26
ours-large‡	✓	91.9	**92.3**	**87.3/87.4**	**95.2**	**75.7**	**89.8**	**88.59**

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

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
RoBERTa-base‡(Liu et al., 2019b)	90.9	92.3	87.9	85.9
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
MT-DNN-base†(Liu et al., 2019a)	91.1	**94.1**	-	-
SyntaxBERT-base†(Bai et al., 2021)	91.0	92.7	88.5	87.3
ours-base‡	**91.8**	94.0	**89.2**	**88.2**
BERT-large‡(Devlin et al., 2018)	91.0	94.4	91.1	91.5
RoBERTa-large‡(Liu et al., 2019b)	91.2	94.5	91.2	91.9
SyntaxBERT-large†(Bai et al., 2021)	91.3	94.7	91.4	92.1
ours-large‡	**92.1**	**95.5**	**92.9**	**92.8**

4.1 Results

In the experiments, we replace the original attention module with our multiway attention and adaptive fusion mechanism in the BERT model.

Firstly, we fine-tune our model on 6 GLUE datasets. Table 1 shows the performance of our model compared with some other baseline models. It can be seen that the effect of non-pre-trained models is significantly worse than pre-trained models. This is mainly because the pre-trained model has more data from the learning corpus and a powerful information extraction ability. The performance of our proposed mechanism on BERT-base and BERT-large model improves the original BERT models by 2.2% and 2.7%, respectively. These great improvements illustrate our model has improved by three modules: multiway attention, self-interactive augmentation, and normalized aggregation. Moreover, our model also outperforms SyntaxBERT (which is the state-of-the-art external knowledge-based model) by 0.9% on BERT-base and 1.3% on BERT-large, respectively.

Secondly, to verify the overall performance of our method, we also conduct experiments on four other popular datasets. The results are shown in Table 2, our model outperforms vanilla BERT and other competitive models on almost all datasets. In addition, the amount of data in Scitail is relatively small, which makes the variance of the model prediction results larger. However, the proposed method still shows very competitive performance on Scitail, which also shows that the method can make up for the lack of generalization ability with fewer parameters by endowing BERT with subtle difference awareness.

Overall, consistent conclusions can be drawn from these results. Compared with previous work, our method shows very competitive performance in judging semantic similarity, and the experimental results also confirm our idea.

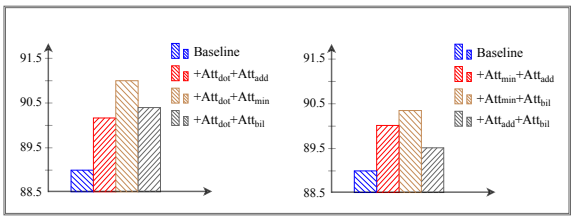

Figure 3: The influence of different attention integration methods on the QQP test set. The baseline model is BERT-base. Att_{dot} indicates Dot Attention, Att_{add} indicates Additive Attention, Att_{min} indicates Minus Attention and Att_{bil} indicates Bilinear Attention.

4.1.1 Ablation Study

To evaluate the contribution of each component in our method, we conduct ablation experiments on the QQP datasets based on BERT. The experimental results are shown in table 3.

The multiway attention module consists of four core components that use different attention functions to model the correlation between sentence pairs. We want to know if each component is useful for the sentence-matching task. First, after removing dot attention, the performance of the model drops by 2.3%. While dot attention can capture the dynamic alignment relationship between word pairs, which is crucial for semantic matching tasks. Then, When additive attention and bilinear attention are removed respectively, the performance of the model on the datasets drops to 90.4% and 90.2%, respectively. They are significantly smaller than Dot attention, which indicates that these two kinds of attention are weaker than dot attention in distinguishing sentence pair relations. Finally, after removing minus attention from the model, the performance dropped by 2.1%. The different 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 when the sub-module is removed, which demonstrates the effectiveness of the internal components of the multiway attention module.

Next, in the aggregation module, we also conduct multiple experiments to verify the effect of augmentation and aggregation of multiple matching features. On the QQP datasets, we first remove the self-interactive augmentation module, and the performance drops to 90.8%. Since self-interactive augmentation can capture interactions between multiple signals, this interaction information is crucial for fusing multi-source vectors. Second, after removing the normalized aggregation module, we only integrate multiple signals by simple averaging. The accuracy drops to 90.5%, which proves that dynamic aggregation according to different weights can further improve the performance of the model. Finally, when we remove the both augmentation module and aggregation module and use simple averaging instead, the performance drops sharply to 89.6%, which is the largest drop among all ablation components. This suggests that while multiway attention mechanisms are crucial for judging sentence pair relations, hard-integrating multi-attention mechanisms without interactive augmentation into PLMs may destroy their pre-existing knowledge, while self-interactive augmentation and soft aggregation can better enhance and aggregate multiple attention.

Table 3: Results of component ablation experiment.

Model	Dev	Test
ours	**92.6**	**92.0**
w/o Dot attention	90.4	89.7
w/o Additive attention	91.1	90.4
w/o Minus attention	90.7	89.9
w/o Bilinear attention	91.3	90.2
w/o Self-interactive augmentation	91.9	90.8
w/o Normalized aggregation	91.6	90.5
w/o Augmentation and Aggregation	90.3	89.6

Finally, to explore whether any two kinds of attention can serve as complementary cues, we aggregate the four kinds of attention in pairs, and the performance improvement over baseline BERT is shown in Figure 3. First, we can find that after the aggregation of different kinds of attention, the performance of the baseline based on a single attention mechanism is improved. Second, The fusion of dot attention and minus attention achieves the best complementary performance improvement among all aggregations, and the combination of dot attention and other attentions significantly outperforms other attention pairs, which reflects that dot attention contributes more to the text matching task than other attentions. Finally, it is worth noting that other types of attention aggregation minus attention can also achieve better results. This may be because minus attention can capture the different information in sentence pairs, and can intuitively reflect the differences between sentence pairs.

Overall, due to the efficient combination of each

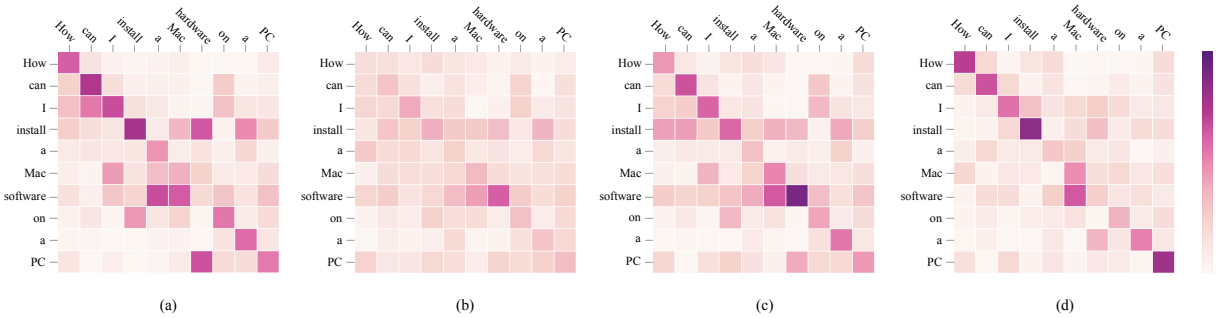

Figure 4: Distribution of Dot attention (a), Additive attention (b), Minus attention (c) and Bilinear attention (d).

component, our model can adaptively fuse vectors generated from different attentions from multiple perspectives into a pre-trained model and leverage its powerful contextual representation to better infer semantics.

4.2 Attention Distribution

To visually demonstrate the impact of different attention functions inside multiway attention on the interactive alignment of sentence pairs, we show the weight distribution of four kinds of attention in the figure 4. We can observe that the word-pair information in the sentence pairs concerned with different attention functions is inconsistent. First, in Figure (a), Dot attention can pay attention to the same words and semantically related words in sentence pairs, but it is heavily influenced by the same words in sentence pairs. It focuses too much on the shallow features of the same text and ignores the deep semantic association of the different words between "software" and "hardware". This shows that using Dot attention alone may lead to wrong predictions. Secondly, in Figure (b), we can see that the distribution of attention weights is more uniform, because the calculation method of additive attention tends to fuse the two signals, so it pays attention to different word pairs of software and hardware to a certain extent, but the interaction weights still small. Next, in Figure (c), it can be observed that Minus attention explicitly pays attention to the difference between "software" and "hardware", and its attention weight is the largest among all word pairs. This is because subtractive attention uses element-wise subtraction to compare the differences between sentence pairs. The greater the difference between word pairs, the greater their weight. Therefore, it can also be complementary to Dot attention. Finally, in Figure (d), the attention weights in bilinear attention focus on the same words, which indicates that bilinear attention tends to focus on the same parts of sentence pairs, and this mechanism is beneficial for capturing sentence pairs' commonality. In summary, different attention focus on different word pairs in sentence pairs. Intuitively, multiway attention can effectively combine the alignment relationships of multiple views in sentence pairs to generate vectors that better describe the matching details of sentence pairs.

The more result of our additional experiments is shown in Appendix A.

5 Conclusion

In this paper, we propose a novel Multiway Attention and Adaptive Fusion mechanism to improve the large-scale pre-trained model, such as BERT, and RoBERTa for semantic matching. The proposed mechanism fully exploits complementary and complex relationships between sentences compared to the single attention-based model. Moreover, the self-interactive augmentation enables better interaction between each attention function and its input, enhancing the representation ability of each attention mechanism. Furthermore, the proposed adaptive aggregation module with normalized aggregation mechanisms effectively fuses the key features and filters out the unrelated features produced by the multiway attention module for semantic understanding. Extensive experiments on 6 GLUE benchmark datasets, as well as 4 other commonly used semantic understanding datasets, verify that our model achieves remarkable performance improvements over the original single attention mechanism-based BERT model as well as other state-of-the-art semantic understanding models. Since the mechanism is a universal transformation mechanism for transformers, it is expected to be applied to other large-scale pre-trained models in the future.

Limitations

This work has the following limitations: (1) The proposed method is based on the introduction of multiple attention functions. Since the introduced attention functions have not been pre-trained, if they are not fine-tuned on the labeled dataset, errors may be introduced and propagated to the decision model, resulting in label prediction errors. (2) We initially demonstrated that external structures can be combined with BERT to improve performance on various SSM tasks. We are also interested in trying to combine it with other PLMs. However, due to computational resource constraints, we did not conduct more experiments on other PLMs. (3) Introducing an extrinsic structure significantly improves the generalization ability of PLMs in few-shot scenarios, but a deeper understanding of why this is the case is still lacking. This may inspire better methods to exploit pre-trained models.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

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*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Self-attention attribution: Interpreting information interactions inside transformer. *arXiv preprint arXiv:2004.11207*, 2.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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*.

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.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: getting inside bert's linguistic knowledge. *arXiv preprint arXiv:1906.01698*.

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.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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

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*.

Jian Song, Di Liang, Rumei Li, Yuntao Li, Sirui Wang, Minlong Peng, Wei Wu, and Yongxin Yu. 2022. Improving semantic matching through dependency-enhanced pre-trained model with adaptive fusion. *arXiv preprint arXiv:2210.08471*.

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

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*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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, 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.

Sirui Wang, Di Liang, Jian Song, Yuntao Li, and Wei Wu. 2022. Dabert: Dual attention enhanced bert for semantic matching. *arXiv preprint arXiv:2210.03454*.

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

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.

Chao Xue, Di Liang, Pengfei Wang, and Jing Zhang. 2024. Question calibration and multi-hop modeling for temporal question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19332–19340.

Chao Xue, Di Liang, Sirui Wang, Jing Zhang, and Wei Wu. 2023. Dual path modeling for semantic matching by perceiving subtle conflicts. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

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.

Yicheng Zou, Hongwei Liu, Tao Gui, Junzhe Wang, Qi Zhang, Meng Tang, Haixiang Li, and Daniel Wang. 2022. Divide and conquer: Text semantic matching with disentangled keywords and intents. *arXiv preprint arXiv:2203.02898*.

A Appendix

A.1 The Result of Additional Experiments

A.1.1 Vs LLMs

We also compared the semantic understanding ability of the Large Language Models(LLMs) with our model. Considering that the answers provided by the Large Language Models(LLMs) involve content analysis and generation, in order to make the comparison results more comparative, we performed the following operations on the experiment. For the convenience of statistics, we have extracted partial data randomly from several datasets as test data. In addition, We have made the following settings for the prompt: Please provide the similarity between the following two sentences. If similar, provide 1; if dissimilar, provide 0.

From the table 4, we can see that although LLMs have strong comprehension and generation abilities, there are still some shortcomings in dealing with semantic matching tasks, mainly due to the inability to understand prompts well. For example, for some sentence pairs, providing some unwanted answers cannot directly provide 1 or 0, especially in large-scale data processing, which can generate many answers that cannot be programmed and batch processed in downstream tasks. In addition, there may be some deviations in the understanding of sentences, leading to incorrect judgments. Moreover, some sentence pairs generate very unstable answers. For example, after the model's judgment, sometimes it gives a result of 1, but when executed again later, the model gives a result of 0. These also indicate the instability of large language models in semantic matching tasks.

Table 4: The performance comparison of ours-large with LLaMa family and GPT3 on several datasets. including SNLI, Scitail(Sci), SICK, and TwitterURL(Twi).

Model	SNLI	Sci	SICK	Twi
LLaMA-7B‡(Touvron et al., 2023)	73.1	75.2	68.4	64.7
LLaMa-13B‡(Touvron et al., 2023)	78.5	83.2	80.6	83.1
GPT3‡(Brown et al., 2020)	84.3	90.1	88.7	85.8
ours-large‡	**92.1**	**95.5**	**92.9**	**92.8**

A.1.2 Stability Analysis

We also performed extensive experiments on QQP, SNLI, QNLI, and Scitail datasets to explore the stability of our method. To minimize the impact of randomness in BERT's training, performance levels were averaged over 10 different runs on the development set. The performance distribution box diagram is shown in Figure 6. The median and average levels of our model exceeded the ordinary BERT on all four datasets, and the performance fluctuation range of our method was within $\pm$ 1% of the average level, which indicates that our method has better stability than BERT on different data distributions.

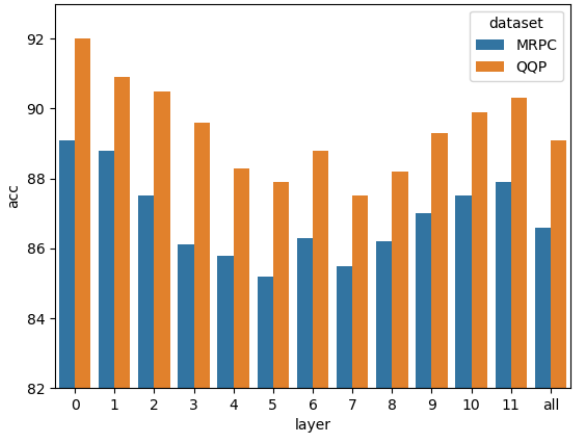

Figure 5: layer-by-layer experiments on MRPC and QQP datasets

A.2 layer-by-layer Analysis

Regarding why we only applied multiway attention to the first layer transformer in BERT, in addition to being inspired by the article in UERBERT (Xia et al., 2021), we also made changes in each layer and also in all layers, as shown in the figure 5. We applied multiway attention to the attention of transformers in each layer of BERT, and at the end, we also made changes to transformers in all layers of BERT, selecting the MRPC and QQP datasets, The experimental results are basically consistent with the conclusion in (Xia et al., 2021), that is, the information understanding capability of the first layer transformer in BERT is the worst, and modifying the multiway attention in this layer is the most effective, better than applying multiway attention in all layers, and also better than applying multiway attention in all layers.

A.2.1 Robustness Performance Test

To examine the performance of our proposed method and competition models in terms of their ability to capture subtle differences in sentence pairs. We performed robustness tests on three widely studied datasets. Table 5 lists the accuracy of the 6 models on the three data sets. We can observe that SwapAnt leads to maximum performance degradation, which indicates that the model

Table 5: The robustness experiments results of our method 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), SwapNameEnt(SN), SwapSyn-WordNet(SW)

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
DistilBERT†(Sanh et al., 2019)	42.24	56.85	83.10	84.09	83.20	-	-	-	-	-
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
SyntaxBERT†(Bai et al., 2021)	49.30	56.37	86.43	84.62	84.19	78.63	95.31	86.91	58.26	76.90
ours‡	**55.93**	**63.26**	**87.75**	85.08	87.99	**81.56**	**97.35**	**85.64**	**60.62**	**81.23**

Model	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
SyntaxBERT†(Bai et al., 2021)	54.92/54.63	53.54/54.73	80.01/79.71	75.46/74.93	57.11/51.95	78.57/79.31
ours‡	**60.75/59.82**	**58.33/60.83**	83.63/83.59	**78.24/78.36**	**60.77/60.35**	**82.58/83.21**

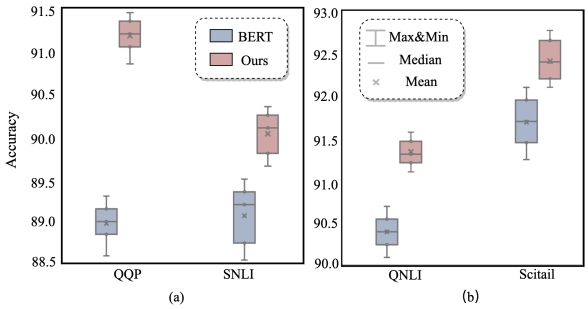

Figure 6: Stability experiments on QQP (a), SNLI (a), QNLI (b), Scitail(b) datasets.

cannot handle the semantic contradiction expressed by antonyms (non-explicit negations) between sentence pairs. The model performance on NumWord drops to 63.26% in Quora datasets, because it requires the model to capture subtle numerical differences for correct language reasoning. Meanwhile, ESIM performed worst. The results reflect that the pre-trained mechanism benefits from abundant external resources and provides better generalization ability than the denovo training model. The performance of the improved pre-trained model SyntaxBERT is better than that of the original BERT model, which reflects that sufficient pre-trained corpus and appropriate external knowledge fusion strategy are helpful to improve the generalization performance of the model. On TwitterType and AddPunc, the performance of our method is lower than that of AlBERT but still better than that of BERT, which may be related to the pre-trained corpus and training mechanism. In the other 8 conversions, our method shows attention to subtle differences and obtains better performance.

A.3 Model Parameter Analysis

Our model is based on the fusion of four different kinds of attention, meanwhile, attention modules are often used to explore the interpretability of the model (Clark et al., 2019; Hao et al., 2020; Lin et al., 2019), in order to prove the impact of the increase of parameter quantity on the model effect, and the impact of multiple fusion of a single attention and the effect of multiple fusion models of different attention, as shown in Table 6, we have conducted experiments on four separate attention on the QQP datasets, and for these four separate attention, 4 times the number of attention are used to participate in the calculation to achieve the same level of parameter quantity as our method. The experimental results show that:

- ◇ The effect of a single kind of attention model is basically the same as that of a model with a single kind of attention expanded by four times, which means that the increase of the number of parameters of the model caused by a simple four-fold increase in the number of single attention does not have a significant improvement on the performance of the model.
- ◇ Our method compared to the model with a single kind of attention expanded by four times have a significant improvement of the result, This shows that the effective fusion of different attachments

does have complementary advantages in semantic extraction.

Table 6: Results of model parameter experiment.

Model	Dev	Test
ours	**92.6**	**92.0**
Dot attention × 1	90.4	89.1
Dot attention × 4	91.2	89.5
Additive attention × 1	90.7	88.9
Additive attention × 4	91.8	89.3
Minus attention × 1	91.9	89.9
Minus attention × 4	90.1	90.2
Bilinear attention × 1	89.6	88.8
Bilinear attention × 4	90.3	89.2

A.4 Case Study

In order to intuitively understand how our method works, we use the three cases in Table 7 for qualitative analysis. First, although S1 and S2 are literally similar in the first example, they express two completely different semantics due to the subtle difference the phrases bring to "eat fruit" and "eat early". The non-pre-trained language model ESIM is difficult to capture the semantic conflict caused by the different words in case 1, so ESIM gives wrong prediction results. The pre-trained language model BERT can identify semantic differences in case 1 and give correct predictions with the help of strong contextual representation capabilities. It is worth noting that the similarity of BERT's predicted sentence pairs is 46.32%, while that of our method is only 1.87%. Second, in case 2, "from 70 to 60" and "from 60 to 50" in sentence pairs express different semantics, but they are mainly caused by numerical differences. Although BERT identified the correct label in case 1 by a small margin, in case 2 it was unable to capture numerically induced differences and gave wrong predictions because it requires the model to capture subtle numerical differences for correct language reasoning. Finally, our model made correct predictions in all of the above cases. Since our models sentence pairs from multiple perspectives, it can pay attention to the small differences in sentence pairs, and adaptively aggregate multi-source information in the aggregation module to better identify the semantics within sentence pairs' differences. At the same time, we can observe that ESIM performs the worst, and the results reflect that the pre-trained mechanism benefits from abundant external resources and provides better generalization ability than the denovo training model. And our BERT-based improved model outperforms the original BERT model, reflecting that a reasonable structure improvement and an effective aggregation strategy can further improve the model's generalization performance.

A.5 Implementation Details of Our Experiments

Implementation Details Our method is based on BERT-base and BERT-large. We set the number of both self-attention layers and heads as 12, and the dimension of embedding vectors as 768. The total number of trainable parameters of both the original BERT-base and our proposed model is the same $110M$ For distinct targets, besides, our hyper-parameters are different. We use AdamW(Loshchilov and Hutter, 2017) 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(He et al., 2016) 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.

A.6 Datasets Statistics

The statistics of all 10 datasets are shown in Table 8.

A.6.1 GLUE Datasets

We experimented with 6 datasets of the GLUE[1] datasets (Wang et al., 2018): MRPC, QQP, STS-B, MNLI-m/mm, QNLI, and RTE, The following is a detailed introduction to these datasets.

- **MRPC** is a dataset that automatically extracts sentence pairs from online news sources and manually annotates whether the sentences in sentence pairs are semantically equivalent. The task is to determine whether there are two categories of interpretation: interpretation or not interpretation.
- **QQP** comes from the famous community Q&A website quora. Its goal is to predict which of the

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

Case	ESIM	BERT	ours
S1: Can eat only fruit for dinner lead to weight loss?	similarity:92.76%	similarity:46.32%	similarity:1.87%
S2: Does ate dinner earlier in the evening help with weight loss?	label:1	label:0	label:0
S1: How do girls lose weight from 70 to 60 ?	similarity:99.51%	similarity:72.66%	similarity:12.06%
S2: How should I lose weight from 60 to 50 ?	label:1	label:1	label:0
S1: What skills do I need to learn to be a successful hardware engineer?	similarity:99.99%	similarity:99.26%	similarity:18.63%
S2: What should I do to become a successful software engineer?	label:1	label:1	label:0

Table 7: The example sentence pairs of our cases. Red and Blue are different phrases in sentence pair.

provided question pairs contains two questions with the same meaning.

- **STS-B** is a collection of sentence pairs extracted from news headlines, video titles, image titles, and natural language inference data. Each pair is annotated by humans, and its similarity score is 0-5. The task is to predict these similarity scores, which is essentially a regression problem, but it can still be classified into five text classification tasks of sentence pairs.
- **MNLI-m/mm** is a crowd-sourced collection of sentence pairs annotated with textual entailment information. Given the promise statement and hypothesis statement, the task is to predict whether the premise statement contains assumptions (entailment), conflicts with assumptions (contradiction), or neither (neutral).
- **QNLI** is a question and answer data set composed of a question paragraph pair, in which the paragraph is from Wikipedia, and a sentence in the paragraph contains the answer to the question. The task is to judge whether the question and sentence (sentence, a sentence in a Wikipedia paragraph) contain, contain and do not contain, and classify them.
- **RTE** is a series of datasets from the annual text implication challenge. These data samples are constructed from news and Wikipedia. All these data are converted into two categories. For the data of three categories, neutral and contradiction are converted into not implication in order to maintain consistency.

A.6.2 Other Datasets

We also experimented with 4 other popular datasets: SNLI[2], Scitail[3], SICK[4] and TwitterURL[5]. The following is an introduction to these 4 datasets.

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
STS-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 8: The statistics of all 10 datasets.

- **SNLI**(Bowman et al., 2015) is a dataset used for classification (or natural language inference). The task is to determine whether two sequences entail, contradict or are mutually neutral.
- **Scitail**(Khot et al., 2018) is an entailment dataset created from multiple-choice science exams and web sentences. Each question and the correct answer choice are converted into an assertive statement to form the hypothesis.
- **SICK**(Marelli et al., 2014) is a dataset for semantic textual similarity estimation. The task is to assign a similarity score to each sentence pair.
- **TwitterURL**(Lan et al., 2017) is a collection of sentence-level paraphrases from Twitter by linking tweets through shared URLs. Its goal is to discriminate between duplicates and not.

A.7 Baselines

To evaluate the effectiveness of our proposed method 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 PLMs (Radford et al., 2018; Devlin et al., 2018) for comparison. Moreover, we also selected several competitive non-pre-

[2]https://nlp.stanford.edu/projects/snli/
[3]https://allenai.org/data/scitail
[4]http://marcobaroni.org/composes/sick.html
[5]https://github.com/lanwuwei/Twitter-URL-Corpus

trained models 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). Besides, We also compared with Large Language Models, such as GPT3, and the LLaMa family.

- **BIMPM** is proposed in (Wang et al., 2017) proposes a Bilateral Multi-Perspective Matching (Bilateral Multi-Perspective Matching, BiMPM) model for sentence matching.
- **ESIM** (Chen et al., 2016) is a model that combines BiLSTM and attention proving that the sequential inference model based on chained LSSM can outperform previous complex structures. It further achieved new SOTA performances.
- **CSRAN** (Tay et al., 2018) is a deep architecture, involving stacked recurrent encoders. CSRAN incorporates two novel components to take advantage of the stacked architecture. It first introduces a new bidirectional alignment mechanism that learns affinity weights by fusing sequence pairs across stacked hierarchies. And then it leverages a multi-level attention refinement component between stacked recurrent layers.
- **Transformer** (Vaswani et al., 2017) uses the attention mechanism to reduce the distance between any two positions in the sequence to a constant. It is not a sequential structure similar to RNN, so it has better parallelism.
- **ELMO** (Peters et al., 2018) adopts a typical two-stage process. The first stage is pre-training using a language model; The second stage is to extract the word embedding of each layer of the network corresponding to the word from the pre-training network and add it to the downstream task as a new feature. It can solve the problem of polysemy of the previous language model because the generated word vector is changed according to the change of the specific use context.
- **GPT** (Radford et al., 2018) is a semi-supervised learning method that uses a large amount of unlabeled data to let the model learn "common sense" to alleviate the problem of insufficient labeled information. The specific method is to pre-train the model Pretrain with unlabeled data before training Fine-tune for labeled data, and ensure that the two kinds of training have the same network structure.
- **BERT** (Devlin et al., 2018) is a pre-trained language model based on the Transformer architecture. Given that our model implements based on BERT, we naturally compare it with vanilla BERT without prior knowledge.
- **UERBERT** (Xia et al., 2021) conducted lots of experiments to analyze which kind of external knowledge BERT has already known, and directly injected the synonym knowledge into BERT without fine-tuning.
- **SemBERT**(Zhang et al., 2020) incorporates contextual semantics from pre-trained semantic role labeling and is capable of explicitly absorbing contextual semantics over a BERT backbone. SemBERT keeps the convenient usability of its BERT precursor in a light fine-tuning way without substantial task-specific modifications.
- **Syntax-BERT** (Liu et al., 2020) is a framework that integrates the syntax trees into transformer-based models. Unlike us, it explicitly injected syntactic knowledge into checkpoints of models.
- **MT-DNN**(Liu et al., 2019a) not only leverages large amounts of cross-task data but also benefits from a regularization effect that leads to more general representations to help adapt to new tasks and domains. MT-DNN extends the model by incorporating a pre-trained bidirectional transformer language model.
- **GPT3** (Brown et al., 2020) As the successor of GPT-2, GPT-3 has become the largest language model, further expanding the parameter space (175 billion vs. 1.5 billion) and data size (45 TB vs. 40 GB). The model requires no fine-tuning to formulate downstream tasks and has excellent performance in zero-shot and few-shot settings. Based on the multi-task generalization capabilities of GPT-2, GPT-3 has achieved good results on many new tasks, including mathematical addition, news article generation, vocabulary interpretation, and code writing. As the number of parameters increases, the model becomes more powerful.
- **LLaMa**(Touvron et al., 2023) trained on a new mix of publicly available data. LLaMa is a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. Testing conducted to date has been in English and has not and could not cover all scenarios. Therefore, before deploying any applications of LLaMa, developers should perform safety testing and tuning tailored to their specific applications of the model to facilitate the safe deployment of LLaMa.