ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240926

Comateformer: Combined Attention Transformer for Semantic Sentence Matching

Bo Li[1,3,*], **Di Liang**[2] **and Zixin Zhang**[1,3]

[1]Tsinghua University, Beijing, China
[2]Fudan University, Shanghai, China
[3]Baidu Inc., Beijing, China

Abstract. The Transformer-based model have made significant strides in semantic matching tasks by capturing connections between phrase pairs. However, to assess the relevance of sentence pairs, it is insufficient to just examine the general similarity between the sentences. It is crucial to also consider the tiny subtleties that differentiate them from each other. Regrettably, attention softmax operations in transformers tend to miss these subtle differences. To this end, in this work, we propose a novel semantic sentence matching model named ***Com***bined ***Atte***ntion Network based on Trans***former*** model (**Comateformer**). In Comateformer model, we design a novel transformer-based quasi-attention mechanism with compositional properties. Unlike traditional attention mechanisms that merely adjust the weights of input tokens, our proposed method learns how to combine, subtract, or resize specific vectors when building a representation. Moreover, our proposed approach builds on the intuition of similarity and dissimilarity (negative affinity) when calculating dual affinity scores. This allows for a more meaningful representation of relationships between sentences. To evaluate the performance of our proposed model, we conducted extensive experiments on ten public real-world datasets and robustness testing. Experimental results show that our method achieves consistent improvements.

1 Introduction

Semantic sentence matching (SSM) is a core method used in the field of natural language processing (NLP) with the objective of comparing and discerning the semantic correlation between two given phrases. In paraphrase identification [33], SSM is used to determine whether two sentences are paraphrase or not. In natural language inference task [2] also known as recognizing textual entailment, SSM determines whether a hypothesis sentence can reasonably be inferred from a given premise sentence. In the answer sentence selection task [47], SSM is employed to assess the relevance between query-answer pairs and rank all candidate answers. In large language models such as GPT [36] and LLaMA [42], SSM can be used for parallel corpus alignment and data denoising. However, the task of establishing the logical and semantic link between two statements is not straightforward, mostly because of the challenge posed by the semantic gap [15].

Across the rich history of semantic sentence matching research, there have been two main streams of studies for solving this problem. The first is representation based method which encodes each of the sentences and obtain their representation vectors in low-dimensional latent space and then utilize the parameterized matching function for the final matching scores, which focuses on how to get good sentence representations [38, 47, 52]. Another type of semantic matching model is interaction-based, which directly align the sentences based on the attention mechanism and aggregate the matching score to directly make the final decision which focuses on how to align the word pairs in the sentence pair [3, 23, 40, 22, 48]. Recently large-scale pre-trained language models such as BERT [7], RoBERTa [27], are becoming more popular in multiple NLP tasks. Because of their high efficiency and effectiveness in contextual information modeling and sentence level encoding, pre-trained models are also wildly used in semantic matching tasks and achieve significant improvement.Recent work attempts to integrate external knowledge [56, 50, 1] into PLMs. Meanwhile, leveraging external knowledge to enhance PLMs has been proven to be highly useful for multiple NLP tasks [18]. Recent work also attempts to enhance the performance of BERT by injecting knowledge into it, such as SemBERT [56], UER-BERT [50], Syntax-BERT [1], DABERT [48] and so on.

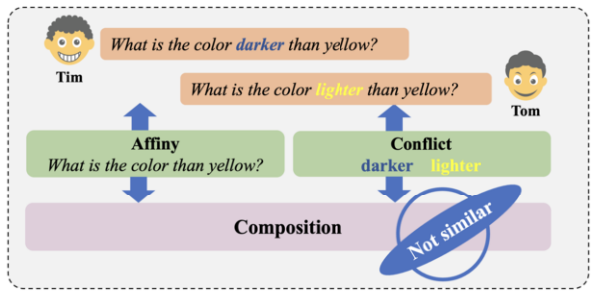


Figure 1. The Combined Attention Network Example for Semantic Sentence Matching.

Although previous studies have provided some insights, those models (e.g., BERT, RoBERTa) do not perform well in distinguishing sentence pairs with high literal similarities but different semantics. Figure 1 exemplifies an instance that is afflicted by this issue. 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. One significant factor is that while the model possesses the capability to assess the level of similarity in overall semantics, it fails to account for the nuanced distinctions present within individual texts. Because for text pairs with highly similar matching words, the overall semantic difference is often caused by different

* Corresponding Author.

local differences. Furthermore, an obvious feature of the attention model is that it can learn relative importance, that is, assign different weights to input values, and the Softmax operator is its core. Softmax makes the weight of core words higher and the weight of non-core words lower. Sparsegen [35] 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.

In this paper, we propose a novel approach named Comateformer, designed exclusively for semantic matching tasks. It replaces vanilla attention in the transformer with combined attention, which works similarly to vanilla attention, but with several key fundamental differences. First, instead of learning relative importance (a weighted sum), combinatorial attention learns combinations of tokens that decide whether to add, subtract, or scale inputs. In other words, our method removes softmax operation because it deviates from the original motivation of attention, so we refer to our method as combined attention. Second, we introduce a quadratically scaled attention matrix, ultimately learning a multiplicative combination of similarity and dissimilarity. We hypothesize that a more flexible design can lead to more expressive and robust models, leading to better performance. To achieve this, we propose two modules to implement the above description. The first is the dual-affinity module, which introduces a negative affinity matrix N in addition to the original affinity matrix E, and the affinity matrix E is obtained by labeling the attention formula, that is, $e_{ij} = a_i \cdot b_j$. In contrast to E, the negative affinity matrix N learns a dissimilarity metric ($n_{ij} = a_i - b_j$) for modeling the differences between word pairs. Subsequently, we introduce a combination mechanism that combines $tanh(E)$ and $sigmoid(N)$ to form a quasi-attention matrix M. In this case, the first term $tanh(E)$ controls the addition and subtraction of vectors, while the auxiliary Affinity N can be interpreted as a gating mechanism that scales unnecessary features when needed. We conduct a series of experiments on 10 datasets and the experimental results show that the method achieves consistent improvements.

The main contributions of this paper are as follows.

- First, we conduct a comprehensive study of the subtle differences between sentence pairs and propose a new method named Comateformer for semantic matching tasks, which has two distinct kinds of functions to represent the interaction between phrase pairs from various viewpoints,and the softmax function was eliminated from the attention mechanism, resulting in an increased receptive field and enhanced capacity to catch tiny differences.
- Secondly, we explicitly integrate Comateformer into both pre-trained and non-pre-trained models, and the results showed that the proposed method can provide greater expressive power, and it can fully discover the inherent complex relationships between sentence pairs for effective semantic matching.
- Finally, we carry out a series of experiments on 10 matching datasets and robustness testing datasets. Experimental results show that Comateformer has achieved consistent improvements, especially in the robustness test, achieving an average improvement of 5% over BERT. The effectiveness of Comateformer is further supported by a case study and an attention distribution analysis, which illustrate the model's nuanced handling of sentence pair interactions and its ability to focus on both commonalities and differences within the text.

2 Related work

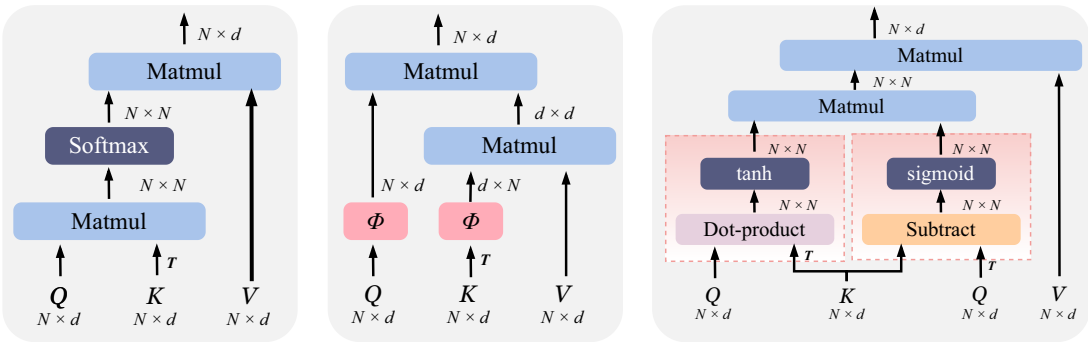
Our work relates to several work in the literature: Semantic Sentence Matching, Robustness test. We will discuss each of these as follows.

2.1 Semantic Sentence Matching

SSM is a focal point within the field of NLP, witnessing significant advancements over the years. It mainly fell into two categories: traditional neural network based methods and pre-trained language model based methods.

Traditional Neural Network based methods. Early approaches to SSM were predominantly reliant on traditional methods such as syntactic features, transformations, and relation extraction [39, 46]. These methods, while effective for specific tasks, were inherently limited in their scope and generalizability. With the advent of large-scale annotated datasets [2, 57] and the proliferation of deep learning algorithms, neural network models have made great progress in SSM. The incorporation of attention mechanisms marked a pivotal shift, offering richer information for sentence matching by elucidating alignment and dependency relationships between sentences [6, 5, 11]. These mechanisms endowed models with the ability to capture nuanced semantic similarities beyond the lexical surface. Concurrently, joint methods that leveraged cross-features through attention mechanisms were introduced to address the limitations of sentence-encoding methods, enhancing performance by capturing word- or phrase-level alignments [49, 10]. The architectural advancements, including the use of residual connections, facilitated the stable increase of network depth, preserving information from lower layers [13]. [25, 37] emphasis on the sequential information and the semantic interdependence of sequences. [51, 29] used distinct convolutional filters to capture the local context. By supplying alignment and dependence relationships between two sentences, the well-established attention processes provided greater information for sentence matching. This was accomplished by giving the information. [4] used an attention method to extract the salient components inside sentences, record the semantic connections, and appropriately align the pieces of two phrases. [28, 31] employed a stacked multi-layer Bi-LSTM with Alignment Factorization to quantify the various levels of features between two texts. Convolutional Neural Network (CNN) focus on the local context extraction with different kernels, and Recurrent Neural Networks (RNN) are mainly utilized to capture the sequential information and semantic dependency. [54] utilized a multi-layer encoding technique and fusion block based on a CNN structure to construct a rapid and highly effective phrase matching model. [8, 9] utilized GNN to leverage the structural information of input sentences in order to achieve full sentence connection modeling.

Pre-trained Language Model based methods. Recently, the pre-trained language models, most notably BERT [7], revolutionized SSM by providing powerful sentence representations through self-supervised learning on vast corpora. This paradigm shift allowed for transfer learning across various NLP tasks, significantly accelerating research progress. One way to enhance the performance of pre-trained models is by modifying the input encoding and utilizing self-supervised pre-trained tasks. XLNet [55] utilized a recently developed PLM task to reduce the disparity between pre-trained tasks



(a) Softmax Attention (b) Linear Attention (c) Combined Attention

Figure 2. Difference between Softmax attention, Linear attention and Combined attention. Softmax attention computes the similarity between all Q-K pairs. Linear attention applies mapping function $\Phi(\cdot)$ to Q and K respectively. Our Combined Attention models both global affinity and local difference information, thus achieving dual perception of affinity and non-affinity, with higher fine-grained differentiation advantages.

and subsequent tasks. Moreover, there are other noteworthy advancements in this field, including RoBERTa [27] and CharBERT [32]. [22, 3] utilizes cross-features as an attention module to express the word-level or phrase-level alignments for performance improvements, and aggregates these integrated information to acquire similarity. DenseNet [53] belongs to the joint approaches which utilizes densely-connected recurrent and co-attentive information to enhance representation. Meanwhile, there is a trend to utilize explicit NLP knowledge to improve sentence representation [30]. For example, [43, 26] used the syntactic dependencies to enhance the sentence representations. The NLP knowledge-enhanced matching models have also adapted to the interaction-based models. For example, MIX [20] utilizes POS and named-entity tags as prior features. SemBERT [56] concatenates semantic role annotation to enhance BERT. UERBERT [50] chooses to inject synonym knowledge. SyntaxBERT [1] integrates the syntax tree into transformer-based models.

The above work has achieved significant advancements in sentence semantic matching, which has motivated us to maximize the utilization of sophisticated neural networks and pre-trained techniques for sentence semantic modeling. However, these models possess the capability to assess the level of similarity in overall semantics, it fails to account for the nuanced distinctions present within individual texts. Because for text pairs with highly similar matching words, the overall semantic difference is often caused by different local differences.

2.2 *Robustness Test*

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 [12]. Tiny literal changes may cause misjudgments. Especially in some cases where fine-grained semantic needs to be discriminated. Besides, most of the current work utilizes one single metric to evaluate their model, may overestimate model capability and lack a fine-grained assessment of model robustness. Therefore, recent work starts to focus on robustness research from multiple perspectives. TextFlint incorporates multiple transformations to provide comprehensive robustness analysis. [21] provide an overall benchmark for current work on adversarial attacks. And [24] propose a more comprehensive evaluation system and add more detailed output analysis indicators.

3 Method

3.1 *Task Definition*

In a formal manner, it is possible to describe each instance of sentence pairings as a triple (Q, P, y). Here, Q represents a phrase of length N, denoted as $(q_1, ..., q_N)$, P represents another sentence of length M, denoted as $(p_1, ..., p_M)$, and y ∈ Y is the label representing the relation between Q and P. In the job of identifying paraphrases, Q and P represent two sentences. The variable y is used to denote the outcome, where Y can take the values of either 0 or 1. Specifically, y = 1 indicates that Q and P are paraphrases of each other, whereas y = 0 indicates that they are not paraphrases. In the context of a natural language inference task, the premise sentence is denoted as Q, the hypothesis sentence as P, and the variable y represents the possible outcomes of the task, namely inference, contradiction, or neutral. Inference refers to the situation where P can be logically deduced from Q, contradiction indicates that P cannot be a valid condition for Q, and neutral signifies that P and Q are unrelated to each other.

A comparison between ***Comateformer*** and classical attention is included in Figure 1. It consists of two parts under the combined attention framework. First, we model the interaction of sentence pairs from different perspectives using two different types of functions. Next, we removed the softmax operation in attention, and gave the attention a wider receptive field and a more subtle difference capture ability. Two sentences are input as $\mathbf{A} \in R^{N_a \times d}$ and $\mathbf{B} \in R^{N_b \times d}$, Where N_a, N_b is the length of sequences A and B. They are padded to the same length N by default. And d is the dimension of the input vector, and returns a combination with the same dimension express. Note that the input is generic as it can be applied to interactive attention for dual sequences and self-attention for single sequences. In the case of single-sequence attention, the variables A and B typically denote identical sequences.

3.2 *Dual Affinity Module*

In this module, we design two different functions, affinity function and difference function, to compare the affinity and difference of vectors between two sentences. First, we compute the pairwise affinities between each word in A and B via the dot product:

$$E_{ij} = \alpha \times F_E(a_i)F_E(b_j). \quad (1)$$

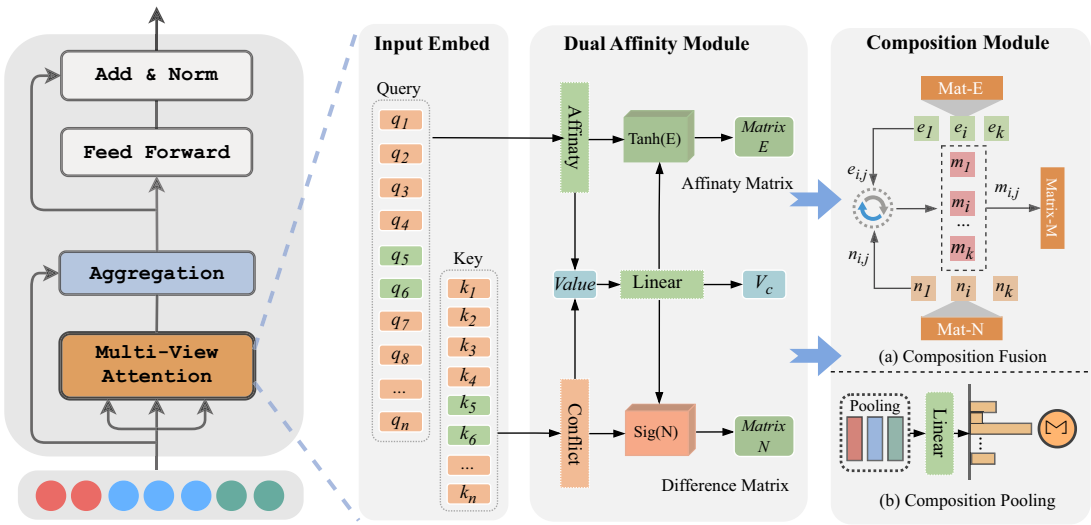


Figure 3. The overall architecture of incorporating Comateformer to transformer Model.

This function computes the pairwise similarity between any two elements in A and B. In this procedure, $F_E(.)$ represents a parameterized function, such as a standard linear/nonlinear function. Additionally, α represents a scaling constant and a non-negative hyperparameter, which can be thought of as a temperature setting that adjusts saturation. Next, as a measure pairwise of negativity (i.e., dissimilarity) between each word in A and B, we perform the following calculation:

$$N_{ij} = \beta \times ||F_N(a_i) - F_N(b_j)||. \tag{2}$$

In this function, we introduce a parameterized function $F_N(.)$ and a scaling constant β, while preserving the L1-Norm l_1. It is noteworthy to mention that in practice we can make parameters shared between $F_E(.)$ and $F_N(.)$. Meanwhile, the affinity matrix N has the same dimensions as the affinity matrix E. Our argument posits that capturing features of different properties (e.g. subtractive compositionality) in attention models is crucial for semantic matching tasks. The fundamental concept underlying negative distance involves utilizing negative affinity values as a gating mechanism to represent negative qualities, a capability that is absent in the original attention method.

3.3 Compositional Attention Module

In the typical vanilla attention, *softmax* is the core component, which is applied to the matrix E to normalize it. Hence, multiplying the normalization matrix of E with the original input sequence yields a vanilla attention pooled representation (aligned representation), where each element in sentence A pools all relevant information for all elements in sentence B. The combined attention we propose is completely different from vanilla attention. First, it has no softmax operation. Specifically, we use the following equations for attention modeling:

$$\mathbf{M} = \tanh(E) \odot sigmoid(N), \tag{3}$$

where M is the final attention matrix in the combined attention mechanism, which is an element-wise multiplication between two matrices.

Normalization of matrix N. Since N is constructed from negative L1 distances, it is clear that *sigmoid(N)* $\in$ [0, 0.5]. Therefore, to ensure that *sigmoid(N)* lies in the range [0, 1], we center the matrix N so that its mean is zero:

$$\mathbf{N} = N - Mean(N). \tag{4}$$

Intuitively, by scaling the matrix N, we preserve the ability to scale up and down the median of the *tanh(E)* matrix, since *sigmoid(N)* has a saturation region between 0 and 1, so it behaves more like a gating mechanism. At the same time we also try the second form of scaling, as an alternative to centering:

$$\mathbf{M} = \tanh(E) \odot (2 * sigmoid(N)). \tag{5}$$

Empirically, we have found that this approach is also very effective.

Temperature. We introduced the hyperparameters α, β that control the size of E and N in the previous sections. Intuitively, these hyperparameters control and affect the temperature of the tanh and sigmoid functions. In other words, high values of α, β will enforce hard-form combined pooling. In this task we set $\alpha = 1$ and $\beta = 1$.

Finally, we apply the Compositional Attention Matrix M to the input sequences A and B with the following formula:

$$\hat{A} = M \times B \quad and \quad \hat{B} = M^T \times A. \tag{6}$$

And the two sentences are update as $\hat{A} \in R^{N_a \times d}$ and $\hat{B} \in R^{N_b \times d}$. Taking $\hat{A}$ as an example, each element A_i in A traverses sentence B and determines whether it contains the token in sentence B by adding (+1), subtracting (-1) or deleting (×0). Similarly, each element in sentence B traverses sentence A and decides to add, subtract, or delete a token from A. Intuitively, which can capture both affinity and dissimilarity features, facilitating rich and expressive representations, unlike typical attention pooling methods that operate on sequences.

3.4 Incorporating Comateformer to Transformer

As shown in Figure 3, which shows the location of Comateformer integrated in the transformer and the schematic diagram of the spe-

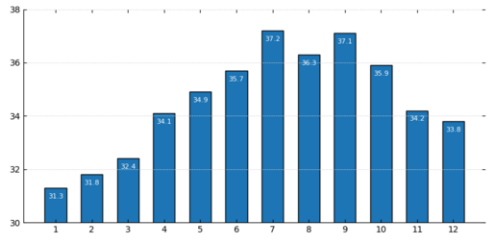


Figure 4. Performance of each BERT layer on TextFlint transformed dataset.

cific modules of Comateformer. The original Transformers [44] employ a self-attention mechanism, which can be interpreted as cross-attention on the same sequence. Our Comateformer replaces the original attention module with de-softmaxed Dual Affinity Module, that is, the original Transformer internal attention equation $A = softmax(\frac{QK^T}{\sqrt{d_k}}) * V$ is now changed to:

$$\mathbf{A} = \tanh(\frac{\mathbf{QK}^T}{\sqrt{d_k}}) \odot sigmoid(\frac{\mathbf{G(QK)}}{\sqrt{d_k}}) * \mathbf{V}, \quad (7)$$

where G(.) is the negation of outer L1 distance between all rows of Q against all rows of K. We either apply centering to $(\frac{G(QK)}{\sqrt{d_k}} * V)$ or 2 * $sigmoid(\frac{G(QK)}{\sqrt{d_k}}) * V$ to ensure the value is in [0, 1]. Finally, both affinity matrices are learned by transforming Q, K, V only once.

3.5 Incorporating Comateformer to PLMs

How to integrate the modified Comateformer with the pre-trained model is also challenging. Injecting additional structure may destroy the representation ability of the pre-trained model. How to gently inject Comateformer into pre-trained models remains a difficult problem. [16] proves that the bottom layer of PLMs pays more attention to words and syntactic information, and the higher layers pay more attention to semantic information. Based on this conclusion, we disassembled the BERT main layer and verified the lack of differential information in different layers of BERT. By solving these problems, we can figure out which layers of BERT are missing differential information. Therefore, we use the robustness testing tool TextFlint as an experimental data set to study the above issues. First, TextFlint makes slight changes to each sampled example so that the sentence pairs have subtle differences. Second, we freeze the parameters of the BERT model (except the softmax classification output head) and adopt pre-trained contextualized word representations for the TextFlint task. This approach allows us to examine the extent to which syntax-related knowledge is stored in each layer of BERT and identify areas lacking this knowledge.

Figure 4 presents the performance of the BERT model layer-by-layer for difference awareness. We leverage TextFlint to perform syntax structure transformations on the dataset, and the performance results are averaged over five different runs. A higher score indicates a stronger proficiency. From the figure, we observe that after freezing the layer parameters of BERT, the sensitivity to difference differs among the layers, with the middle and upper layers being more sensitive to difference than the lower layers. In summary, building upon the insights from the layer-by-layer analysis, we have identified a direction: incorporating Comateformer into the lower layers of BERT. In order to minimize the damage to the original pre-training process, we replace the multi-head attention in the first to third layers with Comateformer in the ratio of 50%, 40%, and 30%.

Table 1. The statistics of all 10 datasets.

Datasets	#Train	#Dev	#Test	#Label	Metrics
MRPC	3669	409	1380	2	Accuracy/F1
QQP	363871	1501	390965	2	Accuracy/F1
MNLI-m/mm	392703	9816/9833	9797/9848	3	Accuracy
QNLI	104744	40432	5464	2	Accuracy
RTE	2491	5462	3001	2	Accuracy
STS-B	5749	1500	1379	2	Pearson/Spearman corr
SNLI	549367	9842	9824	3	Accuracy
SICK	4439	495	4906	3	Accuracy
Scitail	23596	1304	2126	2	Accuracy
TwitterURL	42200	3000	9324	2	Accuracy

4 Experimental Settings

4.1 Datasets

We conduct the experiments to test the performance of Comateformer on 10 large-scale publicly available sentence matching benchmark datasets. The GLUE benchmark [45] is a widely used benchmark test suite in the field of NLP that encompasses various tasks such as sentence pair similarity detection and textual entailment[1]. We have conducted experiments on six sub-datasets of the GLUE benchmark: MRPC, QQP, STS-B, MNLI, RTE, and QNLI. In addition to the GLUE benchmark, we also conduct experiments on four other popular datasets: SNLI [2], SICK [34], TwitterURL [19] and Scitail [17]. The statistics of all 10 datasets are shown in Table 1. Furthermore, to evaluate the robustness of the model, we also utilize the TextFlint [12] tool for robustness testing. TextFlint[2] is a multilingual robustness evaluation tool that tests model performance by making subtle modifications to the input samples.

4.2 Baselines

To evaluate the effectiveness of our proposed Comateformer in SSM, we mainly introduce BERT [7], SemBERT [56], SyntaxBERT, UER-BERT [50] and multiple other PLMs [7] for comparison. In addition, we also select several competitive models without pre-training as baselines, such as ESIM [3], Transformer [44] , etc [14, 49, 41]. In robustness experiments, we compare the performance of BERT on the robustness test datasets. For simplicity, the compared models are not described in detail here.

5 Results and Analysis

5.1 Model Performance

To determine the efficacy of our method, we examine the effectiveness of aggregating Comateformer in 10 datasets, respectively. Table 2 compares the performance of Comateformer and competing models across 10 datasets. It is evident that the performance of non-pre-trained models is considerably inferior to that of pre-trained models. This is mainly because the pre-trained model has more data from learning corpus and powerful information extraction ability. When the backbone model is BERT-base or BERT-large, the average accuracy after integrating Comateformer is improved by 1.1% and 0.8%, respectively. The results show the effectiveness of our Comateformer Model on semantic matching tasks. In addition, our method outperforms RoBERTa-base by 1.6% and RoBERTa-large by 0.6%, respectively. which demonstrates that Comateformer can effectively capture the relationship between sentences from different aspects, so that more fine-grained and complex relationships can

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

Table 2. The performance comparison of Comateformer with other methods.

Model	Pre-train	MRPC	QQP	STS-B	MNLI-m/mm	QNLI	RTE	SNLI	Sci	SICK	Twi	Avg
BiMPM	✗	79.6	85.0	-	72.3/72.1	81.4	56.4	-	-	-	-	-
CAFE	✗	82.4	88.0	-	78.7/77.9	81.5	56.8	88.5	83.3	72.3	-	-
ESIM	✗	80.3	88.2	-	75.8/75.6	80.5	-	88.0	70.6	71.8	-	-
Transformer	✗	81.7	84.4	73.6	72.3/71.4	80.3	58.0	84.6	72.9	70.3	68.8	74.4
BiLSTM+ELMo+Attnt	✓	84.6	86.7	73.3	76.4/76.1	79.8	56.8	89.0	85.8	78.9	81.4	78.9
OpenAI GPT	✓	82.3	81.3	80.0	82.1/81.4	87.4	56.0	88.4	84.8	79.5	81.9	80.4
UERBERT	✓	88.3	90.5	85.1	84.2/83.5	90.6	67.1	90.8	92.2	87.8	86.2	86.0
SemBERT	✓	88.2	90.2	87.3	84.4/84.0	90.9	69.3	90.9	92.5	87.9	86.8	86.5
SyntaxBERT	✓	89.2	89.6	88.1	84.9/84.6	91.1	68.9	91.0	92.7	88.7	87.3	86.3
DABERT	✓	89.1	91.3	88.2	84.9/84.7	91.4	69.5	91.3	93.6	88.6	87.5	86.7
BERT-Base	✓	87.2	89.1	86.8	84.3/83.7	90.4	67.2	90.7	91.8	87.2	84.8	85.8
BERT-Base-Comateformer	✓	**89.3**	**89.6**	**87.3**	**85.2/84.9**	**91.1**	**68.9**	**91.2**	**92.4**	**88.0**	**86.8**	**86.9**
BERT-Large	✓	88.9	89.3	87.6	86.8/86.3	92.7	70.1	91.0	94.4	91.1	91.5	88.0
BERT-Large-Comateformer	✓	**89.7**	**90.4**	**88.1**	**86.9/86.7**	**93.3**	**72.2**	**91.5**	**94.7**	**91.6**	**92.2**	**88.8**
RoBERTa-Base	✓	89.3	89.6	87.4	86.3/86.2	92.2	73.6	90.8	92.3	87.9	85.9	87.6
RoBERTa-Base-Comateformer	✓	**89.8**	**91.1**	**88.4**	**87.5/87.4**	**93.7**	**82.3**	**91.2**	**93.2**	**89.6**	**87.7**	**89.2**
RoBERTa-Large	✓	89.4	89.7	90.2	89.5/89.3	92.7	83.8	91.2	94.3	91.2	91.9	90.3
RoBERTa-Large-Comateformer	✓	**90.3**	**91.4**	**90.9**	**90.1/89.8**	**94.2**	**84.4**	**91.7**	**94.6**	**91.2**	**92.2**	**90.9**

Table 3. Results of ablation experiment of various composition functions.

Model	QQP		QNLI		SNLI	
	Dev	Test	Dev	Test	Dev	Test
Comateformer	**89.8**	**89.6**	**92.2**	**91.1**	**92.2**	**91.1**
$\tanh(\hat{E}) \odot sigmoid(N)$	89.7	89.3	92.4	91.2	92.4	91.2
$\tanh(E) \odot sigmoid(\hat{N})$	89.6	89.5	92.3	91.1	92.3	91.1
$\tanh(E) \odot \tanh(N)$	86.5	85.2	87.3	85.8	87.3	85.8
$\tanh(E) \odot \arctan(N)$	85.1	84.6	86.4	84.3	86.4	84.3
$sigmoid(E) \odot \tanh(N)$	84.8	83.9	85.7	83.8	85.7	83.8
$sigmoid(E) \odot \arctan(N)$	86.2	85.0	87.4	85.6	87.4	85.6
$sigmoid(E) \odot sigmoid(N)$	89.4	87.8	90.7	88.4	90.7	88.4

be exploited. These results demonstrate the advantages of combined attention modeling in mining semantics. Compared with previous work, our method shows very competitive performance levels in evaluating semantic similarity. In addition, the experimental results further verify the effectiveness of our method.

5.2 Ablation study

To assess the individual impact of each component within our methodology, we have performed ablation experiments on the QQP, QNLI and SNLI datasets based on BERT. The experimental findings are shown in Table 3. In this study, we further examine the necessity of centering E and N, and the experimental results on the first two rows of three datasets. It has been discovered that centering matrix E and N does not help performance in most cases. Furthermore, it is seen that applying $Tanh$ on matrix E and $Sigmoid$ on matrix N outperforms other configurations for the proposed attention mechanism. This observation implicitly indicates the efficacy of the combinatorial attention.

5.3 Robustness test performance

We conducted robustness tests on SNLI dataset. Figure 5 lists the accuracy of Comateformer and BERT. We can observe that in SwapAnt our model outperforms BERT nearly 6%, which indicates that Comateformer can better handle semantic contradictions caused by antonyms. And the model performance drops to 77.2% on SwapNum transformation, while Comateformer outperforms BERT by nearly

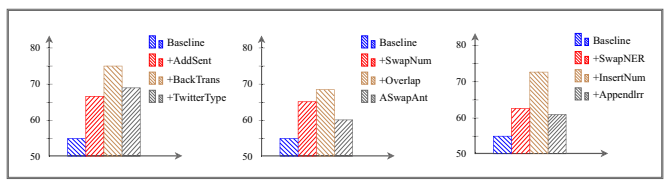


Figure 5. The robustness experiment on the QQP and QNLI datasets based BERT.

5% because it requires the model to capture subtle entity differences for correct linguistic inference. In other transformations, Comateformer still outperforms the baseline, which reflects its effectiveness.

5.4 Case Study

In order to intuitively understand how Comateformer works, we use the three cases in Table 4 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 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 BERT-Comateformer is only 1.87%. Second, in case 2, the sentence pairs "from 70 to 60" and "from 60 to 50" express different semantics, but they are primarily the result of 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 Comateformer 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 alignment module to better identify the semantics within sentence pairs' differences.

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

Case	BERT	BERT-Comateformer
S1: Can eat fruit for dinner lead to weight loss? S2: Does ate dinner earlier help with weight loss?	sim : 46.32% label : 0	sim : 1.87% label : 0
S1: How do girls lose weight from 70 to 60 ? S2: How should I lose weight from 60 to 50 ?	sim : 72.66% label : 1	sim : 12.06% label : 0
S1: What should I learn to be a hardware engineer? S2: What should I learn to be a software engineer?	sim : 99.26% label : 1	sim : 18.63% label : 0

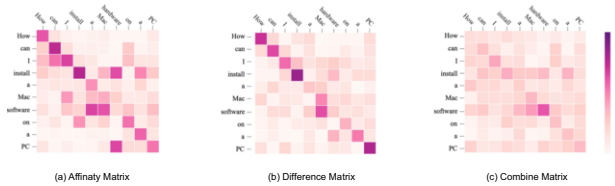


Figure 6. Distribution of Affinaty Matrix (a), Difference Matrix (b), Combine Matrix (c)

5.5 *Attention Distribution*

To visually demonstrate the impact of different attention functions inside multi-channel attention on the interactive alignment of sentence pairs, we show the weight distribution of three kinds of attention in Figure 6. We can observe that the word-pair information in the sentence pairs concerned with different attention functions is inconsistent. First, in Figure 6(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 6(b), 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 minus 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 6(c), the attention weights in combined attention focus on the same and different words, which shows that combined attention can both focus on the same part of the sentence pair and capture different parts, and this mechanism can capture both Affinity and dissimilarity of sentence pairs. In summary, different attention focus on different word pairs in sentence pairs. Intuitively, our method can effectively combine the alignment relationships of multiple perspectives in sentence pairs to generate vectors that better describe the matching details of sentence pairs.

6 Conclusion

In this work, we propose a combination attention network based on transformer model for semantic sentence matching named Comateformer. This model successfully captures the different information that is contained in pairs of words and integrates it into a model that has already been pre-trained. The core of Comateformer lies in its dual-affinity module and compositional attention mechanism, which jointly capture the nuanced similarities and dissimilarities between sentence pairs. This unique capability enables Comateformer to discern subtle semantic differences that often evade traditional attention-based models. The qualitative case study and attention distribution analysis provide clear insights into how Comateformer operates, revealing its ability to adaptively focus on relevant aspects of sentence pairs to enhance semantic understanding. The results of our experiments on 10 publicly available datasets as well as a robustness dataset show that the consistent improvements across various metrics, especially the remarkable gains in robustness testing, underscore the effectiveness of our approach. In future work, we will extend Comateformer to other NLP tasks and develop more sophisticated methods for integrating external knowledge into the model architecture.

References

[1] J. Bai, Y. Wang, Y. Chen, Y. Yang, J. Bai, J. Yu, and Y. Tong. Syntax-bert: Improving pre-trained transformers with syntax trees. *arXiv preprint arXiv:2103.04350*, 2021.

[2] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[3] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

[4] K. Cho, A. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

[5] J. Choi, K. M. Yoo, and S.-g. Lee. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] J. Dong, M.-A. Rondeau, and W. L. Hamilton. Distilling structured knowledge for text-based relational reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6782–6791, 2020.

[9] Z. Fei, Q. Zhang, T. Gui, D. Liang, S. Wang, W. Wu, and X.-J. Huang. Cqg: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6896–6906, 2022.

[10] Y. Gong, H. Luo, and J. Zhang. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*, 2017.

[11] T. Gui, Q. Zhang, J. Gong, M. Peng, D. Liang, K. Ding, and X.-J. Huang. Transferring from formal newswire domain with hypernet for twitter pos tagging. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2540–2549, 2018.

[12] T. Gui, X. Wang, Q. Zhang, Q. Liu, Y. Zou, X. Zhou, R. Zheng, C. Zhang, Q. Wu, J. Ye, et al. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. *arXiv preprint arXiv:2103.11441*, 2021.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[15] J. Im and S. Cho. Distance-based self-attention network for natural language inference. *Cornell University - arXiv,Cornell University - arXiv*, Dec 2017.

[16] G. Jawahar, B. Sagot, and D. Seddah. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[17] T. Khot, A. Sabharwal, and P. Clark. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[18] E. Kiperwasser and M. Ballesteros. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240, 2018.

[19] W. Lan, S. Qiu, H. He, and W. Xu. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*, 2017.

[20] L. Li, Q. Liao, M. Lai, D. Liang, and S. Liang. Local and global: Text matching via syntax graph calibration. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11571–11575. IEEE, 2024.

[21] Z. Li, J. Xu, J. Zeng, L. Li, X. Zheng, Q. Zhang, K.-W. Chang, and C.-J. Hsieh. Searching for an effective defender: Benchmarking defense against adversarial word substitution. *arXiv preprint arXiv:2108.12777*, 2021.

[22] D. Liang, F. Zhang, Q. Zhang, and X.-J. Huang. 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, 2019.

[23] D. Liang, F. Zhang, W. Zhang, Q. Zhang, J. Fu, M. Peng, T. Gui, and X. Huang. 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, 2019.

[24] P. Liu, J. Fu, Y. Xiao, W. Yuan, S. Chang, J. Dai, Y. Liu, Z. Ye, Z.-Y. Dou, and G. Neubig. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*, 2021.

[25] Y. Liu, C. Sun, L. Lin, and X. Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[26] Y. Liu, M. Gardner, and M. Lapata. Structured alignment networks for matching sentences. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1554–1564, 2018.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[28] Y. Liu, M. L. Di Liang, F. Giunchiglia, X. Li, S. Wang, W. Wu, L. Huang, X. Feng, and R. Guan. Local and global: temporal question answering via information fusion. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5141–5149, 2023.

[29] Y. Liu, D. Liang, F. Fang, S. Wang, W. Wu, and R. Jiang. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[30] Y. Liu, M. Li, D. Liang, X. Li, F. Giunchiglia, L. Huang, X. Feng, and R. Guan. Resolving word vagueness with scenario-guided adapter for natural language inference. *arXiv preprint arXiv:2405.12434*, 2024.

[31] R. Ma, Y. Tan, X. Zhou, X. Chen, D. Liang, S. Wang, W. Wu, T. Gui, and Q. Zhang. Searching for optimal subword tokenization in cross-domain ner. *arXiv preprint arXiv:2206.03352*, 2022.

[32] W. Ma, Y. Cui, C. Si, T. Liu, S. Wang, and G. Hu. Charbert: character-aware pre-trained language model. *arXiv preprint arXiv:2011.01513*, 2020.

[33] N. Madnani, J. Tetreault, and M. Chodorow. 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, June 2012. Association for Computational Linguistics. URL https://aclanthology.org/N12-1019.

[34] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli, et al. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik, 2014.

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

[36] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[37] S. Peng, H. Cui, N. Xie, S. Li, J. Zhang, and X. Li. Enhanced-rcnn: an efficient method for learning sentence similarity. In *Proceedings of The Web Conference 2020*, pages 2500–2506, 2020.

[38] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[39] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli. Investigating a generic paraphrase-based approach for relation extraction. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 409–416, 2006.

[40] J. Song, D. Liang, R. Li, Y. Li, S. Wang, M. Peng, W. Wu, and Y. Yu. Improving semantic matching through dependency-enhanced pre-trained model with adaptive fusion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 45–57, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.4.

[41] Y. Tay, L. A. Tuan, and S. C. Hui. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*, 78:154, 2017.

[42] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[43] K. Tymoshenko and A. Moschitti. Cross-pair text representations for answer sentence selection. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2162–2173, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1240. URL https://aclanthology.org/D18-1240.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[45] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[46] M. Wang, N. A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 22–32, 2007.

[47] S. Wang, Y. Lan, Y. Tay, J. Jiang, and J. Liu. 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, 2020.

[48] S. Wang, D. Liang, J. Song, Y. Li, and W. Wu. Dabert: Dual attention enhanced bert for semantic matching. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1645–1654, 2022.

[49] Z. Wang, W. Hamza, and R. Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

[50] T. Xia, Y. Wang, Y. Tian, and Y. Chang. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, pages 2466–2475, 2021.

[51] S. Xu, E. Shijia, and Y. Xiang. Enhanced attentive convolutional neural networks for sentence pair modeling. *Expert Systems with Applications*, 151:113384, 2020.

[52] C. Xue, D. Liang, S. Wang, J. Zhang, and W. Wu. 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, 2023.

[53] C. Xue, D. Liang, P. Wang, and J. Zhang. Question calibration and multi-hop modeling for temporal question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19332–19340, 2024.

[54] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen. Simple and effective text matching with richer alignment features. *arXiv preprint arXiv:1908.00300*, 2019.

[55] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[56] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635, 2020.

[57] R. Zheng, R. Bao, Y. Zhou, D. Liang, S. Wang, W. Wu, T. Gui, Q. Zhang, and X. Huang. Robust lottery tickets for pre-trained language models. *arXiv preprint arXiv:2211.03013*, 2022.