Improving Semantic Matching through Dependency-Enhanced Pre-trained Model with Adaptive Fusion

Jian Song[1]* and **Di Liang**[2]* and **Rumei Li**[2] and **Yuntao Li**[2] and **Sirui Wang**[2]
Minlong Peng[3] and **Wei Wu**[2] and **Yongxin Yu**[1]†
[1]Tianjin University, Tianjin, China [2]Meituan Inc., Beijing, China
[3]Fudan University, Shanghai, China
{songjian799, yyx}@tju.edu.cn
{liangdi04, lirumei, wangsirui, liyuntao, wuwei30}@meituan.com
{mlpeng16}@fudan.edu.cn

Abstract

Transformer-based pre-trained models like BERT have achieved great progress on Semantic Sentence Matching. Meanwhile, dependency prior knowledge has also shown general benefits in multiple NLP tasks. However, how to efficiently integrate dependency prior structure into pre-trained models to better model complex semantic matching relations is still unsettled. In this paper, we propose the **D**ependency-Enhanced **A**daptive **F**usion **A**ttention (**DAFA**), which explicitly introduces dependency structure into pre-trained models and adaptively fuses it with semantic information. Specifically, ***(i)*** DAFA first proposes a structure-sensitive paradigm to construct a dependency matrix for calibrating attention weights. ***(ii)*** It adopts an adaptive fusion module to integrate the obtained dependency information and the original semantic signals. Moreover, DAFA reconstructs the attention calculation flow and provides better interpretability. By applying it on BERT, our method achieves state-of-the-art or competitive performance on 10 public datasets, demonstrating the benefits of adaptively fusing dependency structure in semantic matching task.

1 Introduction

Semantic Sentence Matching (SSM) is a fundamental technology in multiple NLP scenarios. The goal of SSM is to compare two sentences and identify their semantic relationship. It is widely utilized in recommendation systems (Zeng et al., 2021), dialogue systems (Yu et al., 2014), search systems (Li and Xu, 2014), and so on (Gao et al., 2018).

Across the rich history of semantic matching research, there have been two main streams of studies for solving this problem. One is to utilize a sentence encoder to convert sentences into low-dimensional vectors in the latent space, and apply a parameterized function to learn the matching scores between them (Conneau et al., 2017; Reimers and Gurevych, 2019). Another paradigm tends to align phrases and aggregate the integrated information into prediction layer to acquire similarity and make a sentence-level decision (Chen et al., 2016; Tay et al., 2017). After the emergence of large-scale pre-trained language models (PLMs), recent work attempts to integrate external knowledge (Miller, 1995; Bodenreider, 2004) into PLMs. For example, SemBERT (Zhang et al., 2020) concatenates semantic role annotation to enhance BERT. UER-BERT (Xia et al., 2021) chooses to inject synonym knowledge. SyntaxBERT (Bai et al., 2021) integrates the syntax tree into transformer-based models. Meanwhile, leveraging external knowledge to enhance PLMs has been proven to be highly useful for multiple NLP tasks (Kiperwasser and Ballesteros, 2018; Bowman et al., 2016).

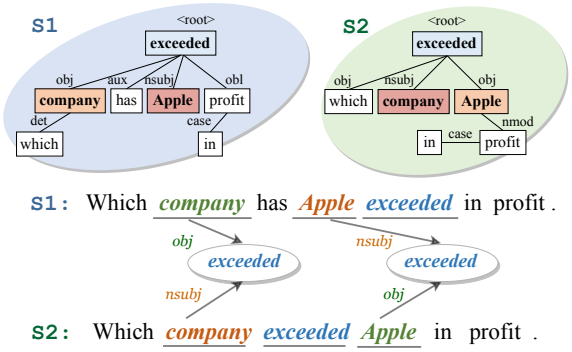


Figure 1: Example sentences that have similar literals, but express different semantics caused by inconsistent dependency.

Although previous work has achieved great progress in SSM, existing models (e.g., *BERT*, *RoBERTa*) still cannot efficiently and explicitly utilize dependency structure to identify semantic differences, especially when two sentences are literally similar. To illustrate that, we display an example of misjudgment by BERT (Devlin et al., 2018) in Figure 1. In the first sentence, the dependency between *"exceeded"* and *"company"* is `obj`, between

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 45 - 57
December 7-11, 2022

"exceeded" and *"Apple"* is nsubj. Its dependency structure is completely opposite to the second sentence. Although the literal similarity of these two sentences is extremely high, the semantics are still quite different. Based on the above observations, we intuitively believe that the dependency structure needs to be considered in the process of semantic matching. From a deeper perspective, the MLM training approach of most existing PLMs is optimizing the co-occurrence probability statistically (Yang et al., 2017), but dependency structure can reflect the dependency relationship within the sentence and integrate prior language knowledge to enhance interaction features. Combined with the actual attention alignment process, we believe that constructing a dependency matrix, ***strengthening the attention weight of same dependency and reducing the attention weight of different dependency***, will further improve the performance of existing PLMs. Therefore, two systemic questions arise naturally:

Q1: How to construct a dependency matrix that contains dependency prior knowledge? Inconsistent dependency structures can lead to severe semantic divergence even between sentences with similar text. To capture the dependency structure features, we propose a structure-aware paradigm to construct the dependency matrix. Our paradigm utilizes three distinct modules, including the dependency similarity between words, the matching of dependency subgraphs, and the *tf-idf* weights.

Q2: How to integrate the introduced dependency signals provided by dependency matrix? To maximize the benefits of the dependency knowledge, we integrate the dependency structure to calibrate our attention alignment. Therefore, our model can not only understand sentence semantics, but also further enhance structural alignment awareness. However, a hard fusion of dependency and semantic signals by simple structure may break the original representing ability of PLMs. How to inject the obtained dependency information softly remains a hard issue. In this paper, we propose an Adaptive Fusion module: ***(i)*** It first inter-aligns these two signals through distinct attentions, and generates vectors describing sentence matching details. ***(ii)*** Then, multiple gates are utilized to extract meaningful information adaptively. ***(iii)*** Moreover, our vectors are further scaled with another fuse-gate to reduce the possibility of noise introduced by dependency features. Eventually, this soft aggregation method can adaptively fuse these collected information and obtain the fusion signals.

Overall, our contributions are mainly as follows:

- We discuss in detail the methodology of further leveraging dependency and explicitly integrating dependency structure into PLMs.
- We propose a novel dependency calibration and fusion network DAFA, which is a meaningful practice combining semantic and dependency information and provides better interpretability. DAFA leverages dependency structure to calibrate attention alignment and constructs a fusion module to adaptively integrate distinct features.
- To verify the effectiveness of DAFA, we conduct extensive experiments on 10 datasets in SSM and achieves state-of-the-art or competitive performance over other strong baselines. It proves the effectiveness of our method.

2 Approach

In this section, we introduce DAFA in detail and the overall architecture is presented in Figure 2.

2.1 How to Build the Dependency Matrix

To construct a dependency matrix that contains dependency knowledge, we propose a structure-aware paradigm with three modules: *(1)* we first use the similarity of dependency trees to build our matrix. *(2)* Then, we introduce subgraph matching to further align the dependency substructures. *(3)* Moreover, we also add *tf-idf* weights to reallocate more attention to keywords and their dependency.

We utilize trigrams to model dependency trees. One trigram unit denotes one branch. In the first sentence of Figure 1, {*"exceeded"*, nsubj, *"Apple"*} is a trigram unit. Apart from literal similarity, two similar trigrams indicate closer semantics. We set n and m as the lengths of sentence A and B. A^i denotes the i-th word of A and $\mathcal{D}_A^i$ denotes the trigram with A^i as the tail node. h_A^i, t_A^i, and r_A^i are indicate the head, tail and type node of $\mathcal{D}_A^i$ respectively. We first utilize the dependency trees similarity to build our matrix $\mathbf{M} \in \mathbb{R}^{n*m}$:

$$\mathbf{M}(i,j) = \underbrace{[s(h_A^i, h_B^j) + s(t_A^i, t_B^j)]}_{\textit{head and tail node}} * \underbrace{r(r_A^i, r_B^j)}_{\textit{type node}}, \quad (1)$$

$$s(a,b) = \{1 \; or \; 0 \,|\, if \; a = b \; or \; \underline{\text{otherwise}}\}, \quad (2)$$

$$r(a,b) = \{\theta \; or \; 1 \,|\, if \; a = b \; or \; \underline{\text{otherwise}}\}, \quad (3)$$

where s, r are binary functions and θ is a con-

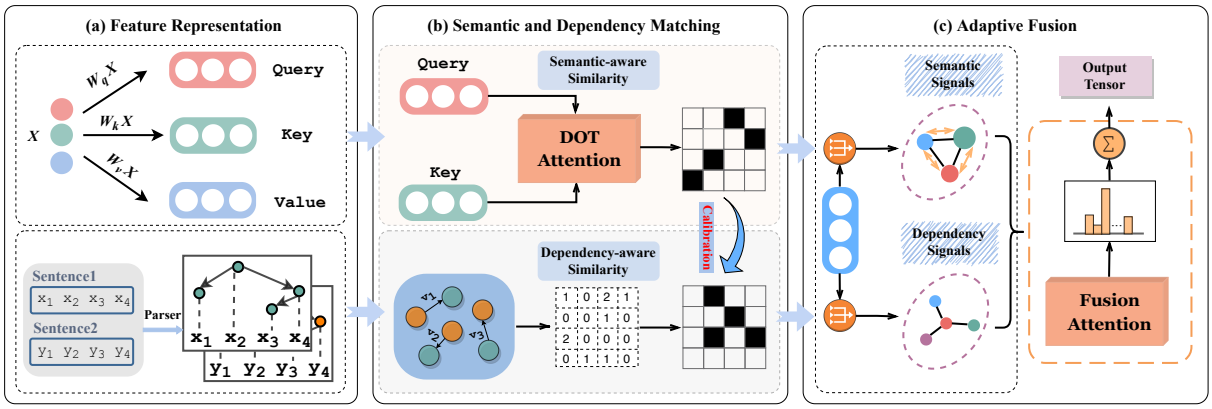


Figure 2: The overall architecture of Dependency-enhanced Adaptive Fusion Attention. It is the structure after we apply DAFA to the original multi-head attention.

stant that determines the effect of dependency type match. However, $\mathbf{M}$ may over-focus on dependency match and neglect the comparison of consecutive dependency trigrams. Therefore, by adopting the subgraph matching mechanism, we can also align the substructures of two dependency trees to acquire the continuous dependency similarity $\mathbf{S}$:

$$\begin{aligned}\mathbf{S}(A^i, B^j) = \{&\alpha s(A^i, B^j) + \nu \mathbf{S}(A^i_x, B^j_y) \mid \\ &\forall A^i_x \in \mathcal{T}^i_A \text{ and } B^j_y \in \mathcal{T}^j_B, \\ &if\ t^i_A = t^j_B \text{ and } r^i_A = r^j_B\}\end{aligned} \tag{4}$$

ν is a decay factor in case $\mathbf{S}$ extremely increase and α is a fixed score to the child nodes of matching trigram pair. $\mathcal{T}^i_A$ denotes the set of child nodes of A^i and A^i_x is the x-th child. $\mathbf{S}$ recursively compares the child nodes of matching pair.

However, our dependency matrix still ignores the difference in importance between words in same sentence. Keywords and their dependency relationship should be allocated more attention. Therefore, we rely on the *tf-idf* weights (Ramos et al., 2003) to obtain the informative scores of distinct sentence components and align the *tf-idf* weights of tail nodes in two trigrams:

$$\mathbf{M_F}(i,j) = \underbrace{|\mathbf{M}(i,j) + \mathbf{S}(A^i, B^j)|}_{\textit{dependency and subgraph}} * \underbrace{(tf_{A^i} * tf_{B^j})}_{\textit{tf-idf weights}} \tag{5}$$

where tf_a denotes the *tf-idf* weight of a. And $\mathbf{M_F} \in \mathbb{R}^{n*m}$ is our final dependency matrix.

2.2 How to Integrate Dependency Information

To better utilize the gained dependency information, we propose to inject our dependency matrix into the original transformer attention module and apply the dependency structure to calibrate the attention alignment. Attention module can be considered as a mapping from query vector $\mathcal{Q}$ and a set of key-value vector pairs ($\mathcal{K}$, $\mathcal{V}$) to the attention distribution. Each layer has multiple parallel attention heads. By introducing $\mathbf{M_F}$, the calculation flow of each head is as follows:

$$\begin{aligned}\mathbf{Sem} &= softmax(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}}) * \mathcal{V}, \\ \mathbf{Dep} &= softmax(\frac{\mathcal{Q}\mathcal{K}^T \odot \mathbf{M_F}}{\sqrt{d_k}}) * \mathcal{V},\end{aligned} \tag{6}$$

where d_k, d_v is the dimension of $\mathcal{K}$, $\mathcal{V}$ and d_{seq} is the input length. We change the dimension of $\mathbf{M_F}$ by adding 1, and ensure each element is in the corresponding position in the sentence alignment. $\odot$ is the element-wise dot product, and $\mathbf{Dep} \in \mathbb{R}^{d_v * d_{seq}}$ denotes the dependency signals from the attention matrix calibrated by our dependency matrix.

However, simple concatenation and fusion seem to underestimate the deep interaction between these two signals and ignore the potential noise introduced by dependency structure. Incorrect structural information may produce noisy outputs and give wrong predictions. Therefore, to further improve the fault tolerance rate of our model and reduce the problem of error propagation, we propose an adaptive fusion module. As shown in Figure 3, ***(i)*** it first interacts and aligns two signals flexibly with semantic-guided attention and dependency-guided attention. ***(ii)*** And then, it adopts multiple gate modules to selectively extract useful features. ***(iii)*** Finally, due to the possibility of noise, a filtration gate is utilized to adaptively filter out inappropriate information.

Firstly, we update the dependency signals through semantic-guided attention. We use $\boldsymbol{s}_i$ and $\boldsymbol{d}_i$ to denote the i-th feature of $\mathbf{Sem}$ and $\mathbf{Dep}$ respectively. We provide each semantic vector $\boldsymbol{s}_i$ to

interact with the dependency signals matrix $\mathbf{Dep}$ and obtain the new dependency feature $\boldsymbol{d}_i^*$. Then, based on $\boldsymbol{d}_i^*$, we can in turn acquire the new semantic feature $\boldsymbol{s}_i^*$ through dependency-guided attention. The calculation process is as follows:

$$\begin{aligned}
\boldsymbol{\delta}_i &= tanh(\mathbf{W}_D\underline{\mathbf{Dep}} \oplus (\mathbf{W}_{s_i}\boldsymbol{s}_i + \boldsymbol{b}_{s_i})),\\
\boldsymbol{d}_i^* &= \underline{\mathbf{Dep}} * softmax(\mathbf{W}_{d_i}\boldsymbol{\delta}_i + \boldsymbol{b}_{d_i}),\\
\boldsymbol{\gamma}_i &= tanh(\mathbf{W}_S\underline{\mathbf{Sem}} \oplus (\mathbf{W}_{d_i^*}\boldsymbol{d}_i^* + \boldsymbol{b}_{d_i^*})),\\
\boldsymbol{s}_i^* &= \underline{\mathbf{Sem}} * softmax(\mathbf{W}_{s_i^*}\boldsymbol{\gamma}_i + \boldsymbol{b}_{s_i^*}),
\end{aligned} \tag{7}$$

where $\mathbf{W}_D, \mathbf{W}_S, \mathbf{W}_{s_i}, \mathbf{W}_{d_i^*} \in \mathbb{R}^{d_{seq}*d_v}$; $\mathbf{W}_{d_i}$, $\mathbf{W}_{s_i^*} \in \mathbb{R}^{1*2d_{seq}}$; $\boldsymbol{b}_{d_i^*}, \boldsymbol{b}_{s_i}, \boldsymbol{b}_{d_i}, \boldsymbol{b}_{s_i^*}$ are weights and bias of our model, and $\oplus$ denotes the concatenation of signal matrix and feature vector.

Secondly, to adaptively capture and fuse useful information from novel semantic and dependency features, we introduce our gated fusion modules:

$$\begin{aligned}
\hat{\boldsymbol{d}}_i &= tanh(\mathbf{W}_{\hat{d}_i}\boldsymbol{d}_i^* + \boldsymbol{b}_{\hat{d}_i}),\\
\hat{\boldsymbol{s}}_i &= tanh(\mathbf{W}_{\hat{s}_i}\boldsymbol{s}_i^* + \boldsymbol{b}_{\hat{s}_i}),\\
g_i &= \sigma(\mathbf{W}_{g_i}(\hat{\boldsymbol{d}}_i \oplus \hat{\boldsymbol{s}}_i)),\\
\boldsymbol{v}_i &= g_i\hat{\boldsymbol{s}}_i + (1-g_i)\hat{\boldsymbol{d}}_i,
\end{aligned} \tag{8}$$

where $\mathbf{W}_{\hat{d}_i}$, $\mathbf{W}_{\hat{s}_i} \in \mathbb{R}^{d_{hid}*d_v}$; $\mathbf{W}_{g_i} \in \mathbb{R}^{1*2d_{hid}}$; $\boldsymbol{b}_{\hat{d}_i}, \boldsymbol{b}_{\hat{s}_i}$ are parameters and d_{hid} is the size of hidden layer. σ is the sigmoid activation and g_i is the gate that determines the transmission of these two distinct representations. By this way, we get the fusion feature $\boldsymbol{v}_i$ that fused the new semantic and dependency signals adaptively.

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

$$\begin{aligned}
f_i &= \sigma(\mathbf{W}_{f_i,s_i}(\boldsymbol{s}_i \oplus (\mathbf{W}_{v_i}\boldsymbol{v}_i + \boldsymbol{b}_{v_i}))),\\
\boldsymbol{l}_i &= f_i * tanh(\mathbf{W}_{l_i}\boldsymbol{v}_i + \boldsymbol{b}_{l_i}),
\end{aligned} \tag{9}$$

where $\mathbf{W}_{f_i,s_i} \in \mathbb{R}^{1*2d_v}$; $\mathbf{W}_{v_i}$, $\mathbf{W}_{l_i} \in \mathbb{R}^{d_v*d_{hid}}$; $\boldsymbol{b}_{v_i}, \boldsymbol{b}_{l_i}$ are trainable parameters and $\boldsymbol{l}_i$ is our final dependency-enhanced semantic feature.

3 Experiments

The datasets, baselines and all details of our experiments are shown in the Appendix B.

3.1 Results

In our experiments, we implemented DAFA in the initial transformer layer of BERT.

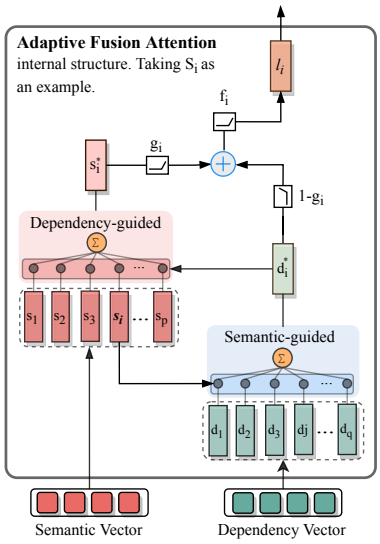


Figure 3: The overall structure of Adaptive Fusion Attention.

The Main Results of GLUE Datasets We first fine-tune our model on 6 GLUE datasets. Table 1 shows the performance of DAFA and other competitive models. It can be seen that using only *non-pretrained models* performs obviously worse than PLMs due to their strong context awareness and data fitting capabilities. When the backbone model is BERT-base and BERT-large, the average accuracy of DAFA respectively improves by **2.23%** and **2.37%** than vanilla BERT. Such great improvement demonstrates the benefit of adaptive fusion dependency structure for mining semantics, and also proves that our framework can help BERT to perform much better in SSM.

Moreover, some previous works such as SemBERT, UERBERT and SyntaxBERT also outperform vanilla BERT by injecting external knowledge, but DAFA still maintains the best performance. Specifically, our model outperforms SyntaxBERT, the top-performing model in previous work leveraging external knowledge, with an average relative improvement of **0.91%** based on BERT-large. Especially on the QQP dataset, the accuracy of DAFA is significantly improved by **2%** over SyntaxBERT. There are two main reasons:

- ◇ On the one hand, we use subgraph matching and keyword weights to enhance the ability of DAFA to capture dependency knowledge. DAFA obtains interactive information that is more conducive to fusing fine-grained features.
- ◇ On the other hand, for the latent noise introduced by external knowledge, our adaptive fusion module can selectively filter out inappropriate signals

Model	Pre-train	Sentence Similarity			Sentence Inference			Avg
		MRPC	QQP	STS-B	MNLI-m/mm	QNLI	RTE	
BiMPM†(Wang et al., 2017)	✗	79.6	85.0	-	72.3/72.1	81.4	56.4	-
CAFE†(Tay et al., 2017)	✗	82.4	88.0	-	78.7/77.9	81.5	56.8	-
ESIM†(Chen et al., 2016)	✗	80.3	88.2	-	-	80.5	-	-
Transformer†(Vaswani et al., 2017)	✗	81.7	84.4	73.6	72.3/71.4	80.3	58.0	74.53
BiLSTM+ELMo+Attn†(Devlin et al., 2018)	✓	84.6	86.7	73.3	76.4/76.1	79.8	56.8	76.24
OpenAI GPT†(Radford et al., 2018)	✓	82.3	70.2	80.0	82.1/81.4	87.4	56.0	77.06
UERBERT‡(Xia et al., 2021)	✓	88.3	90.5	85.1	84.2/83.5	90.6	67.1	84.19
SemBERT†(Zhang et al., 2020)	✓	88.2	90.2	87.3	84.4/84.0	90.9	69.3	84.90
BERT-base‡(Devlin et al., 2018)	✓	87.2	89.0	85.8	84.3/83.7	90.4	66.4	83.83
SyntaxBERT-base†(Bai et al., 2021)	✓	**89.2**	89.6	88.1	**84.9**/84.6	91.1	68.9	85.20
DAFA-base‡	✓	89.0	**91.3**	**89.0**	84.7/**84.8**	**92.3**	**71.3**	**86.06**
BERT-large‡(Devlin et al., 2018)	✓	89.3	89.3	86.5	86.8/85.9	92.7	70.1	85.80
SyntaxBERT-large†(Bai et al., 2021)	✓	**92.0**	89.8	88.5	86.7/86.6	92.8	74.7	87.26
DAFA-large‡	✓	91.6	**91.8**	**89.8**	**87.2/86.9**	**93.7**	**76.2**	**88.17**

Table 1: The performance comparison of DAFA with other methods. We report Accuracy × 100 on 6 GLUE datasets. Methods with † indicate the results from their paper, while methods with ‡ indicate our implementation.

Model	SNLI	Sci	SICK	Twi
ESIM†(Chen et al., 2016)	88.0	70.6	-	-
CAFE†(Tay et al., 2017)	88.5	83.3	72.3	-
CSRAN†(Tay et al., 2018)	88.7	86.7	-	84.0
BERT-base‡(Devlin et al., 2018)	90.7	91.8	87.2	84.8
UERBERT‡(Xia et al., 2021)	90.8	92.2	87.8	86.2
SemBERT†(Zhang et al., 2020)	90.9	92.5	87.9	86.8
SyntaxBERT-base†(Bai et al., 2021)	91.0	92.7	88.1	87.3
MT-DNN-base†(Liu et al., 2019)	91.1	**94.1**	-	-
DAFA-base‡	**91.7**	93.8	**89.8**	**89.4**
BERT-large‡(Devlin et al., 2018)	91.0	94.4	91.1	91.5
SyntaxBERT-large†(Bai et al., 2021)	91.3	94.7	91.4	92.1
MT-DNN-large†(Liu et al., 2019)	91.6	**95.0**	-	-
DAFA-large‡	**92.1**	94.8	**92.4**	**92.8**

Table 2: The performance on 4 other datasets, including SNLI, Scitail(Sci), SICK and TwitterURL(Twi).

to suppress the propagation of noise, while previous work seems to have not paid enough attention to this issue. However, we still notice that SyntaxBERT achieves slightly better accuracy on few datasets. We consider this to be a result of the instability of noise.

The Results of Other Datasets Second, to verify the general performance of our method, we also conduct experiments on 4 other popular datasets. The results are shown in Table 2. DAFA still outperforms vanilla BERT and some representative models on almost all datasets. It is worth noting that although DAFA outperforms MT-DNN (Liu et al., 2019) on SNLI, it does not perform as well as MT-DNN on Scitail. This is because MT-DNN (Liu et al., 2019) uses more model parameters and a large amount of cross-tasks training data, which makes MT-DNN more advantageous in this regard. But MT-DNN will also require more training time and computational cost. Besides, the data volume of Scitail is relatively small, which makes the *variance* of the model prediction results larger. However, DAFA still shows a very competitive performance on Scitail, which also shows from the side that our method can make up for the lack of generalization ability with fewer parameters by endowing BERT with dependency structure awareness.

Overall, consistent conclusions can be drawn from such results. Compared to previous works, our *dependency framework* is highly competitive in further judging semantic similarity, and the experimental results also confirm our thoughts.

3.2 Ablation Study

To assess the contribution of each component in our approach, we have ablation experiments on the QQP dataset based on BERT-large. The experiment results are shown in Table 8.

Dependency Matrix Our *dependency matrix* is jointly constructed by three components: ***(a)*** We first remove the dependency tree similarity and subgraph matching module respectively, and the model performance dropped by nearly **0.7%** and **0.5%**. This suggests that simple dependency structure alignment can further describe the interactions between words to achieve better semantic similarity. ***(b)*** Then, subgraph matching can align the dependency substructures and enrich the contextual representation by introducing finer-grained comparison information. ***(c)*** Besides, due to the different importance of words in the sentence, *tf-idf* can weight each word to readjust the attention distri-

Case	BERT	DAFA-avg	DAFA	Gold
S1:Please help me book a flight from New York to Seattle. S2:Please help me book a flight from Seattle to New York.	label:1	label:0	filter gate:0.93 label:0	label:0
S1:How does reading help you think better? S2:How do you think it's better to read?	label:1	label:0	filter gate:0.91 label:0	label:0
S1:Sorry, I got sick yesterday and couldn't have lunch with you. S2:Sorry, I was taken ill yesterday and unable to meet you for lunch.	label:1	label:0	filter gate:0.15 label:1	label:1
S1:The largest lake in America is in my hometown called Lake Superior. S2:Lake Superior is the largest lake in America, and it's in my hometown.	label:1	label:0	filter gate:0.11 label:1	label:1

Table 7: The example sentence pairs of our cases. Red and Blue are difference phrases in sentence pair. DAFA-avg means replacing the adaptive fusion module with a simple average of semantic and the dependency signals.

Method	QQP	
	DEV	Test
DAFA‡	**92.5**	**91.8**
w/o dependency tree simility‡	91.7	91.1
w/o subgraph matching‡	92.0	91.3
w/o tf-idf weights‡	92.2	91.6
w/o cross attention‡	91.8	91.2
w/o gate mechanism‡	91.9	91.4
w/o adaptive fusion+averaging‡	91.5	91.0

Table 8: The ablation experiment results of our method.

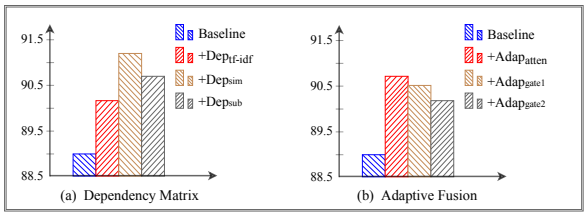


Figure 4: Effects of different sub-module integration methods on the QQP test set. Dep_{sim}, Dep_{sub}, $Dep_{tf\text{-}idf}$ indicate only used sample dependency, subgraphs matching, *tf-idf* weights in dependency matrix respectively. $Adap_{atten}$, $Adap_{gate1}$, $Adap_{gate2}$ represent models that only assemble cross attention, the first gate and the second gate in adaptive fusion respectively.

bution. The accuracy also dropped slightly after the model removed *tf-idf*. The above experiments demonstrate the effectiveness of each component of our *dependency matrix*.

Adaptive Fusion We also conduct multiple experiments to verify the effect of adaptively fusing the original semantic signals and the dependency signals. ***(a)*** We first remove the *cross-attention module*, and the performance drops to **91.2%**. Cross-attention can capture the interaction information between two signals, and interactivity information is crucial for semantic matching. ***(b)*** Moreover, we remove *multiple gate mechanisms*, only relying on the attention modules to integrate our signals. And the accuracy drops to **91.4%**. It shows that the ability of the model to suppress noise is weakened without filter gates. We also replace the overall adaptive fusion module with *simply averaging* and the performance drops sharply to **91.0%**, indicating that soft aggregation and governance can better integrate semantic and dependency signals.

Sub-Module Analysis To further verify the contribution of each submodule to DAFA, we separately assemble the respective sub-modules in dependency matrix computation and adaptive fusion, the results are shown in Figure 4. First, we can find that after adding any of sub-modules, the performance of the model is improved over the baseline. Second, the aggregation of dependency similarity and fusion attention provides the most significant improvements, which intuitively reflects their cornerstone roles as core modules of dependency matrix and adaptive fusion respectively. Such results confirm the necessity of each sub-module again.

Overall, due to the efficient combination of each component, DAFA can adaptively fuse dependency aligned features into pre-trained models and leverage their powerful contextual representation capability to better inference semantics.

3.3 Case Study

To visually demonstrate the validity of our approach, we also conduct a qualitative study using multiple cases in Table 7.

S1 and S2 in the first two cases are literally similar and differ only in the dependency between words, but they express two quite different semantics. In the first example, BERT attempts to capture interaction information from these two sentences, but ignores the dependency between *"New York"* and *"Seattle"*. It fails to distinguish the semantic difference and gives wrong prediction results. By adopting the dependency structure, our method can capture dependency dislocation information and comprehend fine-grained semantics. As the results

show, DAFA gives correct predictions.

However, the injection of dependency structure may generate noise and interference. For example, in the third case, *"got sick"* and *"was token ill"* express the same semantics, but their dependency trees diverge significantly and may mislead the model. By simply averaging the semantic and dependency signals, DAFA-avg instead gives the wrong answers in the last two examples.

Therefore, we propose an adaptive fusion module to reduce the possibility of the noise or useless signals. The filter gate reflects the degree to which the model adopts the dependency structure. In the first two examples, our model learned the important impact of dependency on semantics by adaptively fusing distinct information. DAFA automatically sets the filter gate to 0.91-0.93 and improves perception of dependency structure. However, in the last two examples, our adaptive fusion module identifies the latent noise in dependency structure. To alleviate the possible misleading effect, DAFA correspondingly sets the filter gate to about 0.1, which weakens the model's sensitivity to dependency.

Eventually, as the results show, our DAFA makes correct predictions in all of the above cases and increases the fault tolerance of the model. Such results again demonstrate the effectiveness and necessity of our components.

4 Qualitative Analysis

Attention Distribution and Interpretability To verify the calibration effect of the dependency structure alignment as well as to perform a visual interpretability analysis, we display the attention distribution between two sentences that are literally similar but semantically distinct. The weights for one of our attention heads are shown in Figure 5.

Obviously, vanilla BERT is heavily influenced by the same words in sentences. It ignores deep semantic associations and instead over-focuses on shallow literal features, which may lead to wrong predictions. However, after being calibrated by our method, the attention weights not only learn the shared word in sentences, but also pay more attention to the alignment between the dependency structures. For example, DAFA not only increases the weight of the same *"exceeded"* in the two sentences, but also increases the weight between *"Apple"* in the first sentence and *"company"* in the second one. This is because *"Apple-exceeded"* and *"company-exceeded"* are subject-predicate dependency structures in the two sentences respectively.

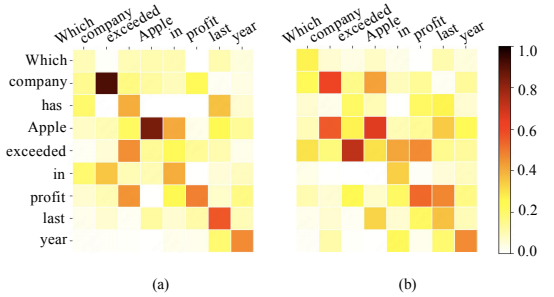


Figure 5: The attention weight distribution of BERT (a) and our method (b).

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 Figure 5(a), we can observe that it is difficult to determine the deep interaction information between sentences by simple contextual features. However, dependency can align sentence structure at the word level. In Figure 5(b), DAFA significantly mitigates the strong influence of the same word and enhances the sensitivity to dependency structures. The calibrated attention distribution is more in line with human cognition and validates our methodology. Since dependency and semantics are linguistic expressions with different perspectives and granularities, their combination can further improve the model's awareness to discern subtle semantic differences and reduce error propagation problems. In addition, the results of ablation experiments also confirm DAFA at each component level and provide results consistent with our predictions. Therefore, our method provides better interpretability.

Data Scenarios and Structural Analysis To further verify the generalization ability of our method, we conduct a range of experiments under different data scenarios on SNLI and Scitail. We use 1% to 100% of the train set corpus to fine-tune our model, and then examine it on the test set. As illustrated in Figure 6(a), our approach obviously improves the performance of BERT in data sparsity scenarios (1%) and always surpasses BERT at different amounts of data (from 1% to 100%). This shows that dependency prior knowledge provides highly salient performance gain happens when the train data is few, and further proves that DAFA can effectively enhance BERT on distinct data scenarios.

To explore which layer most requires the dependency structure, we implement DAFA in the initial transformer layer and in all 12 transformer layers of BERT respectively. Our experiments use 1% to 100% data of MRPC and the results are shown

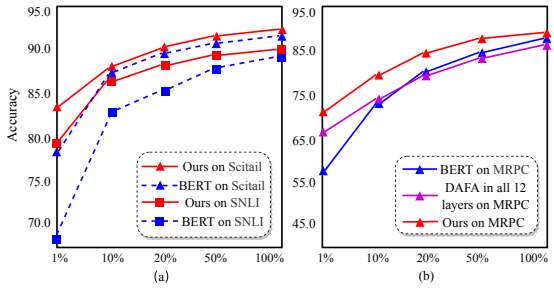


Figure 6: (a) The performance of our model and BERT in different data scenarios of SNLI and Scitail. (b) The performance of implementing DAFA in different layer of BERT on MRPC. The X-axis: the amount of data.

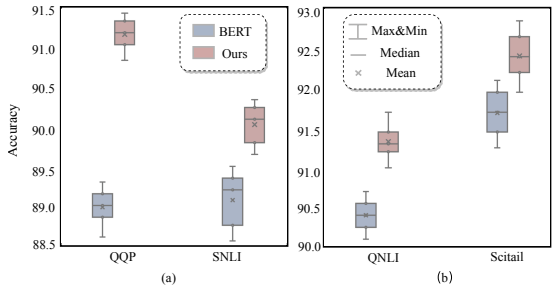


Figure 7: The stability of DAFA and BERT on 4 datasets (QQP, SNLI, QNLI, Scitail).

in Figure 6(b). The effect of our method (initial layer) significantly exceeds vanilla BERT and the approach that implements DAFA in all 12 layers of BERT. The main reason is that BERT pays more attention to word-level features at the bottom layers and semantic features at the top layers.

Stability Study We also conducted extensive experiments on 4 datasets for exploring the stability of our method. To minimize the effect of randomness in BERT training, performance levels are averaged across 10 different runs on the dev set. The performance distribution box plot is shown in Figure 7. The median and mean levels of our model surpass vanilla BERT on all 4 datasets, and the performance fluctuation range of our method is within $\pm 1\%$ around the mean levels, which indicates that our method has better stability relative to BERT on different data distributions.

5 Related Work

Semantic Sentence Matching is a fundamental task in NLP. In recent years, thanks to the appearance of large-scale annotated datasets (Bowman et al., 2015; Williams et al., 2017), neural network models have made great progress in SSM (Qiu and Huang, 2015; Wan et al., 2016), mainly fell into two categories. The first one (Conneau et al., 2017; Choi et al., 2018) focuses on encoding sentences into corresponding vector representations without any cross-interaction and applies a classifier layer to obtain similarity. The second one (Liang et al., 2019a; Chen et al., 2016) utilizes cross-features as an attention module to express the word-level or phrase-level alignments, and aggregates these integrated information to acquire similarity.

Recently, the shift from neural network architecture engineering to large-scale pre-training has achieved outstanding performance in SSM and many other tasks. Meanwhile, leveraging external knowledge (Miller, 1995; Bodenreider, 2004) to enhance PLMs has been proven to be highly useful for multiple NLP tasks (Kiperwasser and Ballesteros, 2018). Therefore, recent work attempts to integrate external knowledge into pre-trained language models, such as AMAN, DABERT, UERBERT, SyntaxBERT, and so on (Liang et al., 2019b; Wang et al., 2022; Xia et al., 2021; Bai et al., 2021).

Dependency Syntax As a crucial prior knowledge, dependency tree provides a form that is able to indicate the existence and type of linguistic dependency relation among words, which has been shown general benefits in various NLP tasks (Bowman et al., 2016). Therefore, many approaches that adopted syntactic dependency information have been proposed and attained multiple great results (Duan et al., 2019). For example, Strubell et al. (2018) present a linguistically-informed self-attention (LISA) in a multi-task learning framework. Sachan et al. (2020) investigate popular strategies for incorporating dependency structure into PLMs. Wang et al. (2022) used a grammar-guided dual-context architecture network (SG-Net) to achieve SOTA effects on span-based answer extraction tasks.

6 Conclusions

In this paper, we propose a Dependency-Enhanced Adaptive Fusion Attention (DAFA), which can adaptively utilize dependency alignment features for semantic matching. Based on the context representation capability of BERT, DAFA enables the model to learn more fine-grained comparison information and enhances the sensitivity of PLMs to the dependency structure. The experiment results on 10 public datasets indicate that our approach can achieve better performance than multiple strong baselines. Since DAFA is an end-to-end trained component, it is expected to be applied to other large-scale pre-trained models in the future.

Limitations

This work has the following limitations: (1) Our proposed method builds on the public part-of-speech and parsing tools. Different parsers may lead to different performances. Meanwhile, errors of these tools may be propagated to our decision model and in turn, result in label prediction error. (2) We initially demonstrated that dependencies can be combined with BERT to improve performance on various SSM tasks. We 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 the dependency prior 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*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

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*.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*.

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

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

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*.

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

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

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Sufeng Duan, Hai Zhao, Junru Zhou, and Rui Wang. 2019. Syntax-aware transformer encoder for neural machine translation. In *2019 International Conference on Asian Language Processing (IALP)*, pages 396–401. IEEE.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

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

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*.

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

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

Hang Li and Jun Xu. 2014. Semantic matching in search. *Foundations and Trends in Information retrieval*, 7(5):343–469.

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. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, page 273.

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.

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

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*.

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

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

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

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

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William Hamilton. 2020. Do syntax trees help pre-trained transformers extract information? *arXiv preprint arXiv:2008.09084*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. *arXiv preprint arXiv:1804.08199*.

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

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

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

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

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

Sirui Wang, Di Liang, Jian Song, Yuntao Li, and Wei Wu. 2022. DABERT: Dual attention enhanced BERT for semantic matching. In *Proceedings of the 29th International Conference on Computational Linguistics*.

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

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

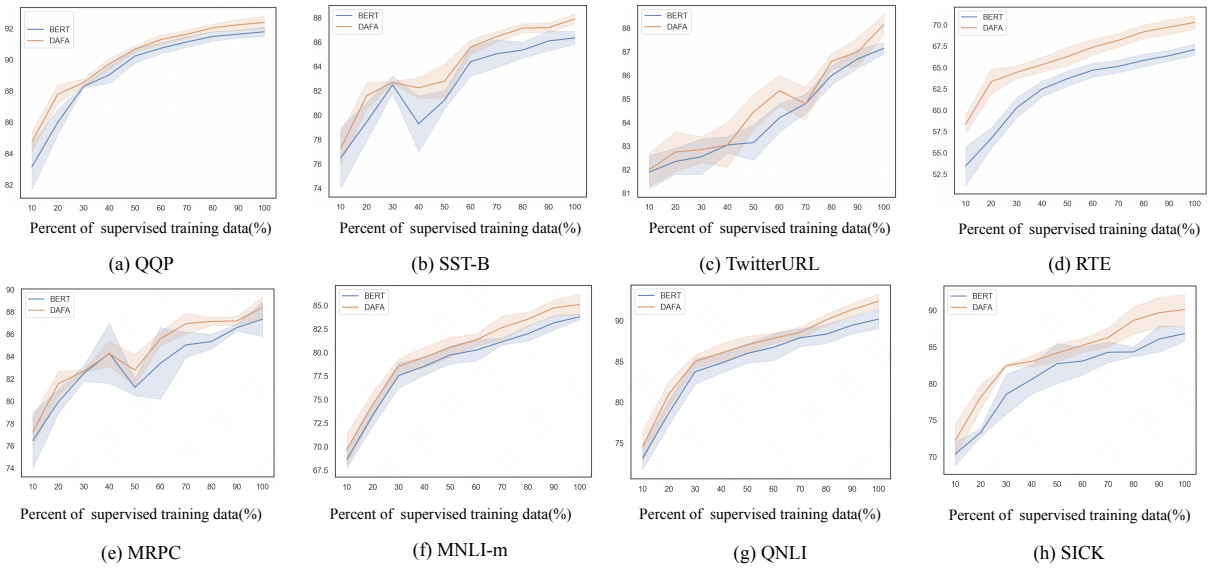
Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2021. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, pages 2466–2475.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. 2017. Breaking the softmax bottleneck: A high-rank rnn language model. *arXiv preprint arXiv:1711.03953*.

Kai Yu, Lu Chen, Bo Chen, Kai Sun, and Su Zhu. 2014. Cognitive technology in task-oriented dialogue systems: Concepts, advances and future. *Chinese Journal of Computers*, 37(18):1–17.

Hansi Zeng, Zhichao Xu, and Qingyao Ai. 2021. A zero attentive relevance matching networkfor review modeling in recommendation system. *arXiv preprint arXiv:2101.06387*.

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.



(a) QQP (b) SST-B (c) TwitterURL (d) RTE

(e) MRPC (f) MNLI-m (g) QNLI (h) SICK

Figure 8: Performance of BERT and DAFA with different amounts of training data. X-axis: Percent of supervised training data. Y-axis: Accuracy of model. The colored bands indicate ± 1 standard deviation corresponding to different percentages of training data.

A Appendix

A.1 Effect of Different Training Data Volumes

We randomly select 10% to 100% of the data from the training data and conduct data scene analysis experiments on eight other datasets. We show the results in Figure 8. For BERT and DAFA, we have trained 5 times for each training scale of each dataset. The changing curves reveal many interesting patterns. First, the performance of our proposed method outperforms vanilla BERT almost uniformly across all training data sizes. Second, on datasets such as RTE, STS-B, and SICK, dependency provide the most significant performance improvement when the training data is small. These findings suggest that if training data is scarce, it is wise to consider injecting dependency knowledge into BERT.

B Appendix

B.1 Implementation Details of Our Experiments

Implementation Details DAFA is based on BERT-base and BERT-large. For distinct targets, our hyper-parameters are different. We use the AdamW optimizer and set the learning rate in $\{1e^{-}5, 2e^{-}5, 3e^{-}5, 8e^{-}6\}$. We set warm-up 0.1, $L2$ weight decay $1e^{-}8$ and constant θ is 2.0. Our epoch is between 3 and 5, and the batch size is selected in {16, 32, 64}. We also set dropout at 0.1-0.3 and gradient clipping in {7.5, 10.0, 15.0}. Our experiments are performed one A100 GPU. Besides, our dependency parser is the biaffine parser proposed by Dozat and Manning (2016). We use original phrase-structure Penn Treebank (PTB) (Marcinkiewicz, 1994) to convert by the Stanford Parser v3.3.02[1] to retrain a parser model. The dependency parser is not updated with our framework.

B.2 Datasets Statistics

The statistics of all 10 datasets is shown in Table 9.

B.2.1 GLUE Datasets

We experimented with 6 datasets of the GLUE[2] 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 extract sentence pairs from online news sources and manually annotate 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 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

[1]https://nlp.stanford.edu/software/lex-parser.shtml
[2]https://huggingface.co/datasets/glue

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.

B.2.2 Other Datasets

We also experimented with 4 other popular datasets :SNLI[3], Scitail[4], SICK[5] and TwitterURL[6]. The following is an introduction to these 4 datasets.

- **SNLI**(Bowman et al., 2015) is a popular dataset used for entailment 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 duplicates or not.

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

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 9: The statistics of all 10 datasets.

B.3 Baselines

To evaluate the effectiveness of our proposed DAFA 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 no pre-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).

- **BIMPM** is propsed in (Wang et al., 2017) and employs a multi-perspective matching mechanism in sentence pair modeling tasks.
- **CAFE** (Tay et al., 2017) introduces a new architecture where alignment pairs are compared, compressed and then propagated to upper layers for enhanced representation learning. And then it adopts factorization layers for efficient and expressive compression of alignment vectors into scalar features, which are then used to augment the base word representations.
- **ESIM** (Chen et al., 2016) proved that the sequential inference model based on chained LSSM can outperform previous complex structures. It futher 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 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) Given that our model implements based on BERT, we naturally compare it with vanilla BERT without prior knowledge. We adopt the configuration of Google's BERT-base in our experiments.
- **UERBERT** (Xia et al., 2021) conducted lots of experiments to analyze which kind of external knowledge that BERT has already known, and directly injected the synonym knowledge into BERT without fine-tuning.
- **SemBERT**(Zhang et al., 2020) incorporates explicit 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 integrate the syntax trees into transformer-based models. Unlike us, it explicitly injected syntactic knowledge into checkpoints of models.
- **MT-DNN**(Liu et al., 2019) 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.