VIRT: Improving Representation-based Text Matching via Virtual Interaction

Dan Li [1]*, **Yang Yang** [2]*, **Hongyin Tang** [2], **Jiahao Liu** [2], **Qifan Wang** [3], **Jingang Wang** [2]†, **Tong Xu** [1]†, **Wei Wu** [2], **Enhong Chen** [1]
[1] University of Science and Technology of China [2] Meituan [3] MetaAI
{lidan528,tongxu,cheneh}@mail.ustc.edu.cn
{yangyang113,tanghongyin,liujiahao12,wangjingang02}@meituan.com
wqfcr@fb.com, wuwei19850318@gmail.com

Abstract

Text matching is a fundamental research problem in natural language understanding. Interaction-based approaches treat the text pair as a single sequence and encode it through cross encoders, while representation-based models encode the text pair independently with siamese or dual encoders. Interaction-based models require dense computations and thus are impractical in real-world applications. Representation-based models have become the mainstream paradigm for efficient text matching. However, these models suffer from severe performance degradation due to the lack of interactions between the pair of texts. To remedy this, we propose a **V**irtual **I**nte**R**ac**T**ion mechanism (VIRT) for improving representation-based text matching while maintaining its efficiency. In particular, we introduce an interactive knowledge distillation module that is only applied during training. It enables deep interaction between texts by effectively transferring knowledge from the interaction-based model. A light interaction strategy is designed to fully leverage the learned interactive knowledge. Experimental results on six text matching benchmarks demonstrate the superior performance of our method over several state-of-the-art representation-based models. We further show that VIRT can be integrated into existing methods as plugins to lift their performances.

1 Introduction

Text matching aims to model the semantic correlation between a pair of texts, which is a fundamental problem in various natural language understanding applications. For instance, in community question answering (CQA) (Zhou et al., 2011; Patra, 2017) systems, a key component is to find similar questions from the database regarding a user question via question matching (Gupta et al., 2018; Sharma et al., 2019). Similarly, a dialogue agent (Welleck et al., 2019) needs to make logical inferences (Conneau et al., 2017; Gao et al., 2021) between a user statement and some pre-defined hypotheses by predicting their entailment relations.

Recently, the wide use of deep pre-trained Transformers (Vaswani et al., 2017) has made remarkable progress in text matching tasks (Raffel et al., 2020a; Ni et al., 2022; Tay et al., 2022).Two paradigms based on fine-tuned Transformer encoders are typically built: interaction-based models and representation-based models, as illustrated in Figure 1(a) & (b). Interaction-based models (e.g., BERT (Devlin et al., 2019)) jointly encode the text pair, which allows the two text sequences to attend each other from the bottom layer to the top layer, resulting in effective matching signals. However, full interaction leads to high computational cost with large inference latency. In addition, text embedding can not be cached or pre-computed, which makes them impractical in many real-world scenarios. For example, in an E-commerce search system, it will cost dozens of days to score millions of query-product pairs with interaction-based models (Chen et al., 2020). Representation-based models (Khattab and Zaharia, 2020; Ni et al., 2022) encode two texts independently with siamese or dual encoders (Cer et al., 2018; Reimers and Gurevych, 2019), which enable the offline-computing of text embeddings and thus significantly reduce the online latency. Unfortunately, independent encoding without any interaction fails to capture the correlation between the text pair, resulting in severe performance degradation.

To balance efficiency and efficacy, several works attempt to equip the siamese structure with late interaction modules. These late interactions are essentially light-weight interaction layers that fuse the two text embeddings from the individual encoders. A variety of late interaction strategies

*Equal contribution.
†Corresponding author.

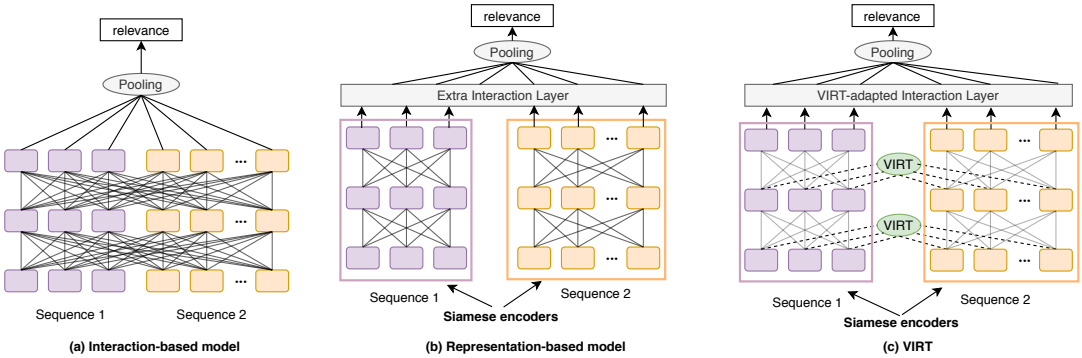


Figure 1: Schematic diagrams illustrating paradigms of text matching. The figure contrasts existing approaches (sub-figures (a) and (b)) with the proposed model (sub-figure (c)).

have been proposed, including MLP layers (Liu et al., 2021), cross-attention layers (Humeau et al., 2020) and Transformer layers (Cao et al., 2020), which obtain considerable improvements on different text matching tasks with reasonable costs. However, these interaction modules are added after Siamese encoders, while interactions in the encoding process of Siamese encoders are still ignored, leaving a large performance gap compared to the interaction-based models.

In this work, we propose a Virtual InteRacTion (VIRT) mechanism with interactive knowledge distillation for improving representation-based text matching while keeping its efficiency. Specifically, Siamese encoders learn interactive information between the pair of texts by mimicking the full interaction, with transferred knowledge from the interaction-based models as guidance. We employ the knowledge transfer as an attention map distillation during training, which is removed during inference to keep the Siamese property, and thus called "virtual interaction". Moreover, we design a VIRT-adapted interaction strategy after Siamese encoding to further leverage the learnt interactive knowledge. Our proposed VIRT is illustrated in Figure 1(c). Experimental results on six text matching benchmarks show the superior performance of VIRT over several state-of-the-art baselines. We summarize the main contributions of this work as follows:

- We propose a novel virtual interaction encoder for representation-based text matching, which effectively models the correlation between a pair of texts without additional inference cost. To the best of our knowledge, it is the first work that introduces interaction into the encoding process of Siamese encoders.
- We develop an interactive knowledge distillation module, which enables deep interaction by transferring knowledge from the interaction-based model. In addition, we design a VIRT-adapted interaction layer to further leverage the learnt interactive knowledge.
- Extensive experiments show that the proposed VIRT outperforms previous SOTA representation-based models, and maintains inference efficiency. The results also indicate that VIRT can be easily integrated into any representation-based text matching models for boosting their performance.

2 Related Work

Text Matching Models Text matching models typically take two textual sequences as input and determine their semantic relationship. Early works perform keyword-based matching such as TF-IDF and BM25 (Pérez-Iglesias et al., 2009). These methods rely on manually defined discrete features, thus usually fail to evaluate the semantic relevance of texts. With the development of deep learning, a large variety of neural models have been proposed for text matching, which use recurrent neural networks (Wu et al., 2017; Mitra et al., 2017; Yang et al., 2016) and convolutional neural networks (Hu et al., 2014) as the backbone, and encode textual sequences into semantic embeddings for fine-grained matches.

Recently Transformer-based models (Bao et al., 2019; Li et al., 2020) leverage self-attention to achieve promising performance on several text matching tasks (Tang et al., 2021; Qu et al., 2021; Xiong et al., 2021). Generally, these models can be classified into interaction-based models

(Logeswaran and Lee, 2018; Devlin et al., 2019) and representation-based models (Reimers and Gurevych, 2019). As a typical interaction-based model, BERT (Devlin et al., 2019) concatenates the text pair as the input and uses its `[CLS]` token embedding to predict the matching (Nogueira and Cho, 2019). In contrast, representation-based models utilize dual encoders to encode the pair of texts individually, which achieve high inference efficiency by pre-computing and storing all text embeddings in the database. However, there is usually a large performance degradation compared to interaction-based models. More recently, late interactions with light attention layers (Humeau et al., 2020; Khattab and Zaharia, 2020; Cao et al., 2020) have been introduced after dual encoders to balance efficiency and efficacy. However, rich interactive information between the text pair is still ignored during encoding.

Knowledge Distillation Knowledge distillation (Hinton et al., 2015; Tang et al., 2019) is to transfer knowledge from a teacher model with better quality to a less complex student model. Various works (Jiao et al., 2020; Sanh et al., 2019; Sun et al., 2019, 2020) have been proposed to compress BERT to a tiny structure with fewer Transformer layers and smaller hidden size through distilling predicted logits and hidden states. There are several recent distillation works that are closely related to our work. DiPair (Chen et al., 2020) performs extra interaction through a light Transformer layer, and distills predicted logits from the interaction-based model. Deformer (Cao et al., 2020) adopts multiple Transformer-based interaction layers and distills the representations as well as the predicted logits from the interaction-based model. However, these methods merely distill logits/representations from interaction-based models to the late interaction layer of representation-based models. In contrast, VIRT distills the attention map from the interaction-based model directly to the encoding process of Siamese encoders, which transfers interactive knowledge more effectively.

3 Methodology

3.1 Preliminaries

Interaction-based Models Given two textual sequences $\boldsymbol{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_m]$ and $\boldsymbol{Y} = [\mathbf{y}_1; \ldots; \mathbf{y}_n]$ as input, the interaction-based models concatenate $\boldsymbol{X}$ and $\boldsymbol{Y}$ into $[\boldsymbol{X}; \boldsymbol{Y}]$, and encode $[\boldsymbol{X}; \boldsymbol{Y}]$ with a Transformer encoder (Devlin et al., 2019): $\mathbf{H}^{(L)} = \text{Enc}([\boldsymbol{X}; \boldsymbol{Y}])$. Each layer of Transformer consists of two residual sub-layers: a multi-head attention operation (MHA) (i.e., Eq. 1a, Eq. 1b) and a feed-forward network (FFN) (i.e., Eq. 1c):

$$\mathbf{M}^{(l)} = \text{softmax}\left(\text{Att}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)})\right), \quad (1a)$$

$$\hat{\mathbf{H}}^{(l)} = \text{LN}\left(\mathbf{M}^{(l)}\mathbf{V}^{(l)} + \mathbf{H}^{(l-1)}\right), \quad (1b)$$

$$\mathbf{H}^{(l)} = \text{LN}\left(\text{FFN}\left(\hat{\mathbf{H}}^{(l)}\right) + \hat{\mathbf{H}}^{(l-1)}\right). \quad (1c)$$

where $\text{Att}(\mathbf{Q}, \mathbf{K}) = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}$ is used to compute the attention map $\mathbf{M}$. d is the dimension of hidden states. $\mathbf{H}^{(l)}$ is the intermediate representation from the l-th layer. $\mathbf{Q} = \mathbf{H}\mathbf{W}_\mathbf{Q}$, $\mathbf{K} = \mathbf{H}\mathbf{W}_\mathbf{K}$ and $\mathbf{V} = \mathbf{H}\mathbf{W}_\mathbf{V}$ are the query, key and value matrices. $\text{LN}(\cdot)$ refers to the Layer-Normalization operation. The interaction-based models are able to encode interactive information into the representations of $\boldsymbol{X}$ and $\boldsymbol{Y}$ through the full attention mechanism.

Representation-based Models In contrast to interaction-based models, representation-based models encode $\boldsymbol{X}$ and $\boldsymbol{Y}$ individually through two independent Siamese Transformer encoders: $\widetilde{\mathbf{H}}_\mathbf{x}^L = \text{Enc}_\mathbf{x}(\boldsymbol{X})$, and $\widetilde{\mathbf{H}}_\mathbf{y}^L = \text{Enc}_\mathbf{y}(\boldsymbol{Y})$. These models are very efficient, especially for downstream retrieval tasks: 1) they do not need to conduct pairwise encoding. 2) text embedding for the corpus can be pre-computed. However, since there is no interaction between $\boldsymbol{X}$ and $\boldsymbol{Y}$ during encoding, fine-grained interactive information would be lost in representation-based models, resulting in significant performance degradation.

3.2 VIRT

The major weakness of representation-based models is lacking interaction when individually encoding two input sequences. Essentially, the interaction-based models perform interaction through the attention mechanism, and compute a unified attention map using both $\boldsymbol{X}$ and $\boldsymbol{Y}$. On the other hand, the representation-based models compute two disjoint attention maps from $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. In the following sections, we first present the details of the difference between these two types of models in terms of the MHA operation. Next, we introduce the VIRT mechanism which improves the representation-based models without extra inference cost.

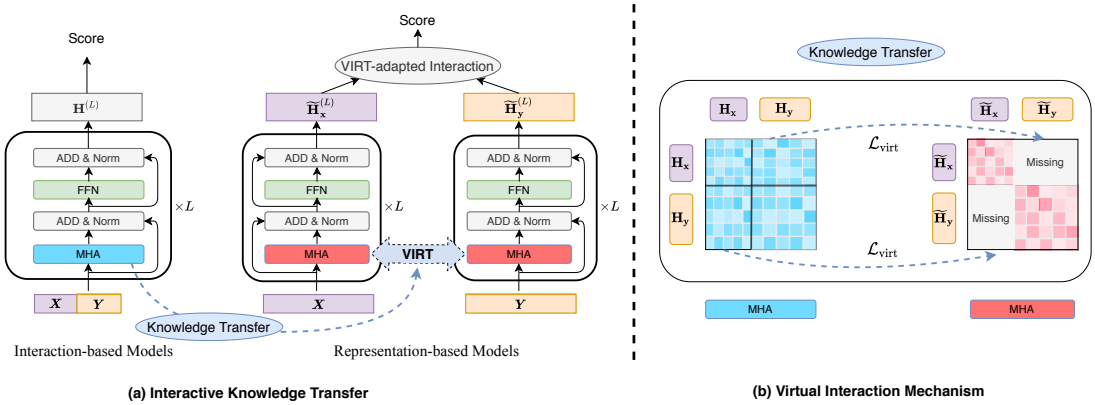


Figure 2: The proposed VIRT model architecture. (a) Interactive knowledge transfer procedure by distilling the attention map from the interaction-based model. (b) VIRT mechanism details.

MHA Analysis The MHA operation in interaction-based models is illustrated by the blue attention map in Figure 2(b). Specifically, the input representations $\mathbf{H}$ of the l-th layer in interaction-based models could be decomposed to the $\boldsymbol{X}$-part and the $\boldsymbol{Y}$-part, i.e., $\mathbf{H} = [\mathbf{H_x}; \mathbf{H_y}]$, where $\mathbf{H_x} = [\mathbf{h}_1; ...; \mathbf{h}_m]$ and $\mathbf{H_y} = [\mathbf{h}_{m+1}; ...; \mathbf{h}_{m+n}]$. Note that we omit the superscript l here for the simplicity of the presentation. In the attention map computation, the query and key matrices could also be rewritten as the combination of the $\boldsymbol{X}$-part and the $\boldsymbol{Y}$-part, i.e., $\mathbf{Q} = [\mathbf{Q_x}; \mathbf{Q_y}]$ and $\mathbf{K} = [\mathbf{K_x}; \mathbf{K_y}]$. According to Eq. 1a, the final attention score before the $\mathrm{softmax}(\cdot)$ operation (denoted as $\mathbf{S}$) could be decomposed as the following partitioned matrix:

$$\begin{aligned}\mathbf{S} &= \mathrm{Att}\left([\mathbf{Q_x}; \mathbf{Q_y}], [\mathbf{K_x}; \mathbf{K_y}]\right) \\ &= \begin{bmatrix} \mathrm{Att}(\mathbf{Q_x}, \mathbf{K_x}) & \mathrm{Att}(\mathbf{Q_x}, \mathbf{K_y}) \\ \mathrm{Att}(\mathbf{Q_y}, \mathbf{K_x}) & \mathrm{Att}(\mathbf{Q_y}, \mathbf{K_y}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_{\mathbf{x}\to\mathbf{x}} & \mathbf{S}_{\mathbf{x}\to\mathbf{y}} \\ \mathbf{S}_{\mathbf{y}\to\mathbf{x}} & \mathbf{S}_{\mathbf{y}\to\mathbf{y}} \end{bmatrix}.\end{aligned} \tag{2}$$

In particular, $\mathbf{S}_{\mathbf{x}\to\mathbf{x}} \in \mathbb{R}^{m\times m}$ and $\mathbf{S}_{\mathbf{y}\to\mathbf{y}} \in \mathbb{R}^{n\times n}$ are the MHA operations performed in $\boldsymbol{X}$ or $\boldsymbol{Y}$ only, which correspond to the MHA operations in representation-based models. $\mathbf{S}_{\mathbf{x}\to\mathbf{y}} \in \mathbb{R}^{m\times n}$ and $\mathbf{S}_{\mathbf{y}\to\mathbf{x}} \in \mathbb{R}^{n\times m}$ represent the interactions between $\boldsymbol{X}$ and $\boldsymbol{Y}$ in interaction-based models, which are responsible for enriching the representations with interactive information. However, these interactions are missing in representation-based models, as illustrated by the missing attention maps in Figure 2(b).

Interactive Knowledge Transfer In order to bring the missing interaction back and bridge the performance gap, we let representation-based models mimic the interactions as:

$$\begin{aligned}\widetilde{\mathbf{M}}_{\mathbf{x}\to\mathbf{y}} &= \mathrm{softmax}\left(\mathrm{Att}(\widetilde{\mathbf{Q}}_{\mathbf{x}}, \widetilde{\mathbf{K}}_{\mathbf{y}})\right), \\ \widetilde{\mathbf{M}}_{\mathbf{y}\to\mathbf{x}} &= \mathrm{softmax}\left(\mathrm{Att}(\widetilde{\mathbf{Q}}_{\mathbf{y}}, \widetilde{\mathbf{K}}_{\mathbf{x}})\right),\end{aligned} \tag{3}$$

where $\widetilde{\mathbf{M}}_{\mathbf{x}\to\mathbf{y}}$ denotes the attention map which is generated by $\widetilde{\mathbf{H}}_{\mathbf{x}}$ attending to $\widetilde{\mathbf{H}}_{\mathbf{y}}$, and similar for $\widetilde{\mathbf{M}}_{\mathbf{y}\to\mathbf{x}}$. These two additional attention maps represent the missing interactive signals in representation-based models, which are responsible for updating the representations. However, they cannot be directly calculated from the dual encoders in representation-based models, resulting in less effective text embeddings.

To close the performance gap between representation-based and interaction-based models, we propose to align the missing attention maps with their counterparts that have already existed in interaction-based models. Intuitively, the attention maps in the interaction-based models can guide the learning of the representations to evolve towards an interaction-rich direction as if the representations have interacted with each other during the encoding process. By this means, we distill the knowledge in interaction and transfer it into the dual encoders without any extra computational cost in inference. That is why we call the mechanism "virtual interaction".

Concretely, we employ a trained interaction-based model as the teacher and distill the knowledge to a representation-based student model. In each layer, we obtain the attention maps $\mathbf{M}_{\mathbf{x}\to\mathbf{y}}$ and $\mathbf{M}_{\mathbf{x}\to\mathbf{y}}$ from the interaction-based model and transfer these supervised interactive knowledge to guide the learning of the representation-based

model. Formally, the goal is to minimize the L_2 distance across all layers between $(\widetilde{\mathbf{M}}_{\mathbf{x}\to\mathbf{y}}, \widetilde{\mathbf{M}}_{\mathbf{y}\to\mathbf{x}})$ and $(\mathbf{M}_{\mathbf{x}\to\mathbf{y}}, \mathbf{M}_{\mathbf{y}\to\mathbf{x}})$:

$$\begin{aligned}\mathcal{L}_{\text{virt}} = \frac{1}{2L}\sum_{l=1}^{L}\Bigg(\frac{1}{m}\left\|\widetilde{\mathbf{M}}^{(l)}_{\mathbf{x}\to\mathbf{y}} - \mathbf{M}^{(l)}_{\mathbf{x}\to\mathbf{y}}\right\|_2 \\ + \frac{1}{n}\left\|\widetilde{\mathbf{M}}^{(l)}_{\mathbf{y}\to\mathbf{x}} - \mathbf{M}^{(l)}_{\mathbf{y}\to\mathbf{x}}\right\|_2\Bigg).\end{aligned} \tag{4}$$

Note that the above distillation is only applied in the training stage to learn better dual encoders. This preserves the Siamese property of representation-based models without extra inference cost.

3.3 VIRT-Adapted Interaction

Through VIRT, interactive knowledge could be incorporated deeply into each encoding layer of the representation-based models. However, after Siamese encoding, the representations of the last layer, i.e., $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}}$ and $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}}$, still cannot see each other, and thus lack explicit interaction. To make full use of the learnt interactive knowledge, we further design a VIRT-adapted interaction strategy, which fuses $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}}$ and $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}}$ under the guidance of the attention map learnt by VIRT.

Specifically, we perform VIRT-adapted interaction between the $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}}$ and $\widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}}$ following the process in Eq.3. The generated attention maps are formulated as follows:

$$\begin{aligned}\hat{\mathbf{M}}^{(L)}_{\mathbf{x}\to\mathbf{y}} &= \text{softmax}\left(\text{Att}(\widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}}, \widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}})\right), \\ \hat{\mathbf{M}}^{(L)}_{\mathbf{y}\to\mathbf{x}} &= \text{softmax}\left(\text{Att}(\widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}}, \widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}})\right), \\ \mathbf{u} &= \text{Pool}\left(\hat{\mathbf{M}}^{(L)}_{\mathbf{x}\to\mathbf{y}}\widetilde{\mathbf{H}}^{(L)}_{\mathbf{y}}\right), \\ \mathbf{v} &= \text{Pool}\left(\hat{\mathbf{M}}^{(L)}_{\mathbf{y}\to\mathbf{x}}\widetilde{\mathbf{H}}^{(L)}_{\mathbf{x}}\right),\end{aligned} \tag{5}$$

where $\text{Pool}(\cdot)$ denotes the mean pooling operation. Eq. 5 employs the same interaction strategy as VIRT, and further utilizes learnt attention maps to update representations explicitly. Finally, we utilize simple fusion to make predictions:

$$\begin{aligned}\mathbf{r} &= (\mathbf{u}, \mathbf{v}, \mathbf{u}-\mathbf{v}, \max(\mathbf{u}, \mathbf{v})), \\ y &= \text{softmax}\left(\text{MLP}(\text{MLP}(\mathbf{r}) + \mathbf{r})\right),\end{aligned} \tag{6}$$

where $(,)$ is the concatenate operation, and MLP denotes the Multi-Layer Perceptron. The overall training objective is minimizing the combination of the task-specific supervision loss $\mathcal{L}_{\text{task}}$ and the distillation loss $\mathcal{L}_{\text{virt}}$:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha\mathcal{L}_{\text{virt}}, \tag{7}$$

where α is a hyper-parameter to weight the influence of virtual interaction. It is noteworthy that VIRT is a general strategy, and can be used to enhance any representation-based matching models, as will be shown in experiments.

4 Experiments

4.1 Datasets

We conduct an extensive set of experiments on three types of datasets, including three sentence-sentence matching tasks (MNLI, QQP, RTE), one question answering task (BoolQ) and two real-world query-passage matching tasks (Q2P, Q2A).

An overview of all the datasets is provided in Table 1. The detailed statistics and average text lengths are presented. Note that the average length of Chinese is based on characters, and English is based on words.

Dataset	# of pairs (Train / Dev)	AvgLen TextA	AvgLen TextB
MNLI	392,702 / 9,815	19.6	10.0
RTE	2,490 / 277	43.0	8.6
QQP	327,464 / 40,430	10.9	11.2
BoolQ	9,427 / 9,427	8.8	92.7
Q2P	110,000 / 13,960	6.0	57.6
Q2A	519,821 / 11,440	3.8	160.0

Table 1: Datasets statistics. (For GLUE and SuperGLUE, the results on development sets are reported since they do not distribute labels for test sets. For Q2P and Q2A datasets, we construct development sets, which is non-overlapping with the training sets.))

MNLI (Williams et al., 2018) is a large-scale entailment classification dataset. The objective is to predict the relationship between a pair of sentences as entailment, neutral, or contradiction.

RTE (Bentivogli et al., 2009) dataset comes from a series of annual competitions on textual entailment. The objective is to predict whether a given hypothesis is entailed by a given premise.

QQP (Sharma et al., 2019) is a large-scale sentence similarity dataset with question pairs from Quora. The task is to determine if the two questions have the same meaning.

BoolQ (Clark et al., 2019) is a question answering dataset for yes/no questions given question and document pairs.

Q2P is a binary classification task derived from the MSMARCO Passage Ranking data (Nguyen

Model	MNLI	RTE	QQP	BoolQ	Q2P	Q2A	Inference Latency (times)
BERT-Base	84.1	66.0	90.6	74.1	91.0	91.0	332.6ms (1.0x)
Siamese BERT (Devlin et al., 2019)	60.2	53.3	80.1	70.5	73.2	80.6	47ms (7.1x)
DeFormer (Cao et al., 2020)	71.1	55.0	88.5	70.9	84.0	84.1	118ms (2.8x)
DiPair (Chen et al., 2020)	71.3	55.1	88.6	71.3	80.3	87.4	49.1ms (6.8x)
Poly-encoder (Humeau et al., 2020)	74.5	57.2	88.5	70.9	83.5	88.3	68.2ms (4.9x)
Sentence-T5 (Ni et al., 2022)	75.9	59.2	90.3	72.0	85.7	81.9	47.5ms (7.0x)
VIRT (ours)	**78.6**	**60.5**	**90.4**	**73.1**	**89.2**	**90.1**	66.5ms (5.0x)

Table 2: Performance comparison on six datasets. Note that we only report online parts of inference latency, since the representation-based embeddings could be computed offline and online latency in real-world scenarios is more concerning. Since models on these six datasets take a similar input setup, we report inference latency on BoolQ and omit the other five. Results are statistically significant with p-value < 0.001.

et al., 2016) containing 110K query passage pairs. Given a (query, passage) pair, the goal is to predict whether the passage contains the answer for the query. The original dataset does not contain labeled negative samples. For each query, we sample the negative passage from the top-100 passages retrieved by BM25.

Q2A is our internal dataset containing a huge amount of query-advertisement pairs. All the data are crawled from a Chinese E-commerce website and manually annotated. Given a (query, advertisement) pair, the goal is to predict the relevance between the advertisement and the query.

4.2 Baselines

We adopt several state-of-the-art representation-based matching models as our baselines.

Siamese BERT (Devlin et al., 2019) is a Siamese architecture that uses pre-trained BERT to separately produce embeddings of two inputs. The pooled output embeddings of two sequences are concatenated to give final predictions.

DeFormer (Cao et al., 2020) is a decomposed BERT-based model, which splits the full self-attention into two independent self-attention in the lower layers of BERT while the upper layers are kept origin with full self-attention.

DiPair (Chen et al., 2020) is a fast and distilled representation-based model for text matching. It performs extra interaction through a light transformer layer, which feeds with truncated embeddings output from the last encoder layer.

Poly-encoders (Humeau et al., 2020) is a representation-based model for pairwise text matching which utilizes an attention mechanism to perform extra interaction after Siamese encoders.

Sentence-T5 (Ni et al., 2022) learns sentence embeddings from text-to-text Transformers T5 (Raffel et al., 2020b). The output embeddings of two sequences and their difference are concatenated to give final predictions.

4.3 Experimental Setup

VIRT setup We use BERT-base (Devlin et al., 2019) as the encoder backbone of VIRT. The parameters are initialized with the pre-trained BERT-base model (uncased). We share all parameters between $\text{Enc}_{\mathbf{x}}(\cdot)$ and $\text{Enc}_{\mathbf{y}}(\cdot)$. We also take BERT-base as the interaction-based model, which is finetuned first, and used as the teacher model to transfer interaction knowledge to representation-based models. The pooling strategy of BERT-base at the prediction layer is fixed to mean pooling (instead of `[CLS]`), as we observe better performance on both BERT-base and all VIRT-enhanced representation-based models.

Implementation Details All baselines are initialized with pre-trained BERT-base parameters, and fine-tuned to achieve the best results on the validation sets. It is worth noting that we fix the total number of transformer layers for all models at 12 to make a fair comparison, though some of the baselines such as DiPair (Chen et al., 2020) take fewer layers for extreme efficiency at the cost of performance. The first 8 and first 16 output token embeddings of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are picked out as DiPair's input, which is the best setting reported from its paper. The number of context vectors in Poly-encoders is 360. For MNLI and QQP, we use the standard partition and metrics on the GLUE benchmark[1]. For RTE and BoolQ, we follow the SuperGLUE[2]. For Q2P and Q2A, we construct the dataset from MSMARCO Passage

[1]https://gluebenchmark.com/

[2]https://super.gluebenchmark.com/

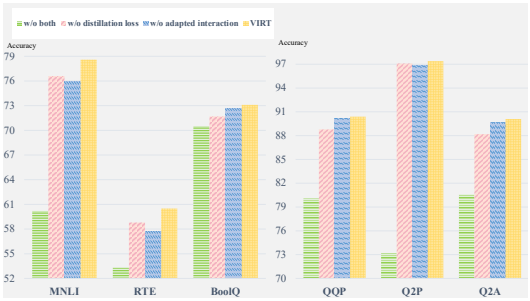


Figure 3: Ablation analysis for different components on all datasets.

Ranking data and real-world E-commerce data using AUC-ROC as the evaluation metric. We split 10% of the training set for tuning hyper-parameters in these tasks, and report results on the original development split.

We implement all models with Tensorflow 1.15 on Tesla V100 GPU (32GB memory). We set α as 1 and the batch size as 28. Training epochs for six tasks are set to 5, 30, 5, 30, 5, 5 respectively. Sequence length of two texts for six tasks are set to $(128, 128)$, $(64, 328)$, $(128, 128)$, $(64, 328)$, $(200, 200)$, $(16, 256)$ respectively. The learning rate is set to $5e-5$, with the warm-up ratio set to 0.1. All models are optimized by Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. For measuring the online inference latency, we run the inference with the batch size set to 28. We repeat each experiment 10 times and report the metrics based on the average over these runs.

4.4 Main Results

The performance comparison of different methods is presented in Table 2. BERT-base shows its effectiveness as a powerful interaction-based model. Siamese BERT has a significant performance decline compared with BERT. DeFormer, DiPair, Poly-encoder and Sentence-T5 achieve considerable improvement compared with Siamese BERT. Finally, VIRT achieves the best performance, outperforming all the representation-based baselines. It even obtains competitive results compared with the interaction-based BERT model. These results validate that VIRT is able to approximate the deep interaction modeling ability of the interaction-based models.

We further compare the inference latency on the BoolQ dataset across different models, which is also listed in Table 2. According to the result, all representation-based models show significant speedup compared with the interaction-based models. The speedup mainly benefits from the Siamese encoder, which enables embeddings computed offline. Siamese BERT achieves the fastest inference speed, yet suffers from a severe performance decline. DeFormer gets relatively higher latency, due to the computation complexity of the extra interaction layers. Dipair truncates the sequence to a shorter length before the interaction layer, which produces an excellent speed-up in terms of online latency. Poly-encoder and Sentence-T5 considerably improve the performance, at the cost of slightly increased computations. Compared with all the baselines, our model shows superiority in terms of performance while keeping the high efficiency at the same time. Note that the inference latency is computed based on the average of all example pairs in an online manner. However, representation-based methods are able to pre-compute the embeddings of the corpus offline, and therefore dramatically reduce the inference time for downstream applications.

4.5 Analysis and Discussion

4.5.1 Ablation Study

To understand the impact of different components in VIRT, we conduct an ablation study by removing each component and retrain the models. In particular, "w/o distillation loss" means removing the optimization goal of Eq. 4. "w/o adapted interaction" means removing the adapted interaction in Eq. 5, and using simple fusion for representation at the last layer as Eq. 6. "w/o both" means remove both strategies simultaneously. The results are shown in Figure 3. The drop in performance without distillation or adapted interaction indicates the effectiveness of these two architectures. For MNLI and RTE, the performance drop caused by removing adapted interaction is more severe. Our hypothesis is that MNLI and RTE are natural language inference tasks, which require more fine-grained matching signals and rely heavily on explicit interaction. For QQP, BoolQ and Q2A, adapted interaction has less effect. However, distillation still brings substantial improvement, which further validates the effectiveness of incorporating interaction.

4.5.2 Layer Importance

In this set of experiments, we apply VIRT to different selected layers in the dual encoder to understand the importance of the interaction

Model	MNLI	RTE	QQP	BoolQ	Q2P	Q2A
DeFormer + VIRT distillation	72.3 (↑1.2)	55.8 (↑0.8)	89.2 (↑0.7)	71.8 (↑0.9)	85.0 (↑1.0)	86.1 (↑2.0)
DiPair + VIRT distillation	71.6 (↑0.3)	55.3 (↑0.2)	88.6 (-0.0)	71.8 (↑0.5)	82.3 (↑2.0)	87.9 (↑0.5)
Poly-encoder + VIRT distillation	75.3 (↑0.8)	57.9 (↑0.7)	89.2 (↑0.7)	71.6 (↑0.7)	84.1 (↑0.6)	89.4 (↑1.1)
Sentence-T5 + VIRT distillation	77.2 (↑2.3)	61.7 (↑2.5)	90.8 (↑0.5)	72.5 (↑0.5)	88.5 (↑2.8)	83.5(↑1.6)

Table 3: Performance gain of applying VIRT distillation to different representation-based models. ↑ represents the performance gain.

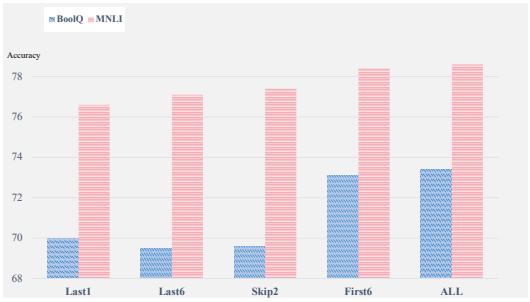


Figure 4: Ablation study of applying VIRT to different encoder layers on MNLI and BoolQ.

knowledge in different encoder layers. (1) VIRT-Last: only applying VIRT to the last k layers. (2) VIRT-First: only applying VIRT to the first k layers. (3) VIRT-Skip: applying VIRT to 1-in-k layers. (4) VIRT-All: applying VIRT to all layers.

The results on MNLI and BoolQ are shown in Figure 4. It is not surprising to see that VIRT-All achieves the best performance over all the compared settings, showing the importance of the interaction for all layers. We observe that VIRT-First performs better than VIRT-Last and VIRT-Skip when all activating 6 layers, which indicates that interaction knowledge from the bottom layers plays a crucial role. We also applied VIRT at the last one layer, referring to (Wang et al., 2020) who claims distilling the last layer is enough. However, we find that when the teacher model and the student model are heterogeneous, merely distilling the information of the last one layer faces great performance degradation.

4.5.3 Impact of VIRT Distillation

To verify the generality and effectiveness of the proposed VIRT distillation, we further import it into the aforementioned representation-based models by applying the knowledge distillation to different baselines. The results are reported in Table 3. According to the results, we can observe that VIRT distillation could be easily integrated into other representation-based text matching models to lift their performances. Note that the results in Table 3 are different from the results of w/o adapted interaction in the ablation study. In the ablation study, we always leverage the fusion layer from Eq. 6, which yields much better performances. Similar observations have been found in Sentence-T5 (Ni et al., 2022).

Model	MNLI
VIRT-BERT-Tiny$_2$	68.1 (↑10.3)
VIRT-BERT-Mini$_4$	70.9 (↑11.8)
VIRT-BERT-Small$_4$	73.6 (↑13.5)
VIRT-BERT-Medium$_8$	74.5 (↑14.3)
VIRT-BERT-Large$_{24}$	79.3 (↑15.4)

Table 4: Performance gain of applying VIRT distillation to models with different configurations.

4.5.4 Different Model Configurations

We apply VIRT (including VIRT distillation and VIRT-adapted interaction) to pre-trained models with different sizes to show its robustness on different numbers of encoder layers. We conduct experiments using BERT-Tiny$(2/128)$, BERT-Mini$(4/256)$, BERT-Small$(4/512)$, BERT-Medium$(8/512)$, BERT-Base$(12/768)$ and BERT-Large$(24/1024)$ on the MNLI dataset, where a/b means the number of encoder layers is a and the dimension of hidden representation is b. The results are reported in Table 4. It can be seen from the results that VIRT yields better performance on all size of the pre-trained models, which is consistent with the observations from the main results.

4.5.5 Impact of α

For our proposed VIRT approach, we conduct additional parameter search over α from $\{0, 0.2, 0.6, 1, 2, 10\}$ in Eq. 7 on the MNLI task. The experimental results are shown in Figure 5. From the results, it is clear that VIRT with $\alpha = 1$ achieves the best performance among all the α values, which illustrated that the $\mathcal{L}_{\text{virt}}$ is as important as $\mathcal{L}_{\text{task}}$. We also observe that the performance of VIRT is relatively stable with a wide range of α, e.g., from 0.6 to 1.

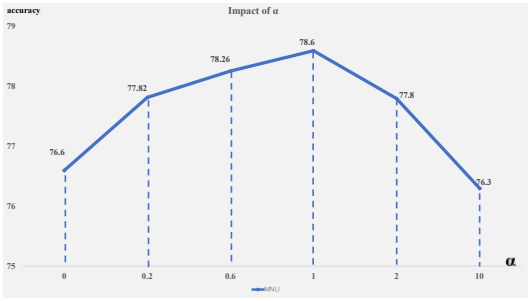
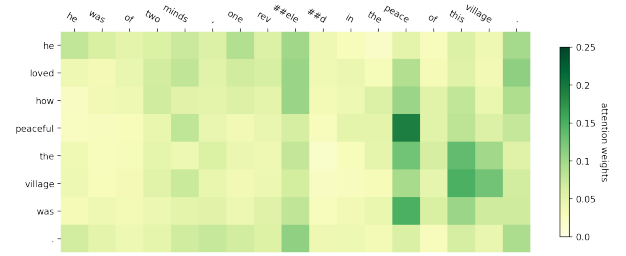


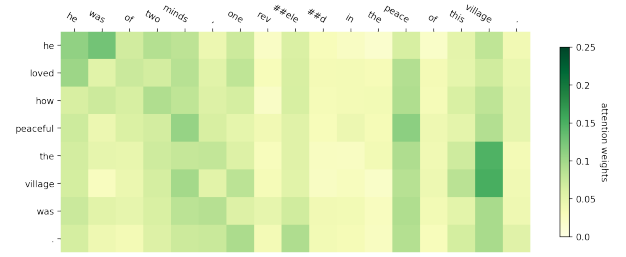
Figure 5: Impact of α on MNLI.

4.6 Case Study

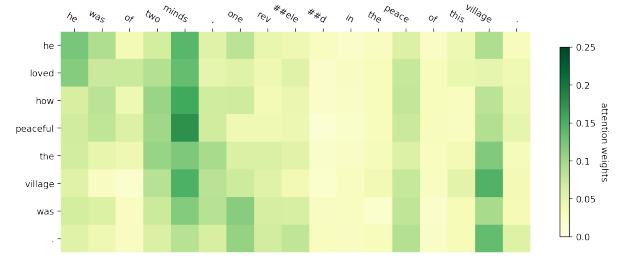
To show the effect of VIRT distillation in a more intuitive way, we visualize the attention matrices of different models. Specifically, we choose an example from the MNLI dataset and plot the corresponding attention matrices of the interaction-based model and the representation-based model with/without VIRT distillation. As shown in Figure 6(a)-6(c), the attention matrix with VIRT distillation is more consistent to the interaction-based model than the model without VIRT. In particular, the interaction-based model aligns "peaceful" with "peace" which can be learnt by VIRT whereas the representation-based model misses this information. As a result, the representation-based model without VIRT fails to predict the two sentences as "neutral" relationship.



(a) The attention matrix of interaction-based model



(b) The attention matrix of representation-based model with VIRT distillation.



(c) The attention matrix of representation-based model without VIRT distillation.

Figure 6: Visualization of the attention matrices.

5 Conclusion

Representation-based models are widely used in text matching tasks due to their high efficiency while under-performing the interaction-based ones caused by lacking interaction. Previous works often introduce extra interaction layers while the interaction in Siamese encoders is still missing. In this paper, we propose a virtual interaction (VIRT) mechanism , which could approximate the interactive modeling ability by distilling the attention map from interaction-based models to the Siamese encoders of representation-based models, with no additional inference cost. The proposed VIRT, which employs knowledge distillation as well as adapted interaction strategy, achieves state-of-the-art performance among existing representation-based models on several text matching tasks.

Limitations

Although the proposed VIRT mechanism enhances the performance of dual encoder architectures and achieves new SOTA on several datasets, two limitations are presented and discussed in this section. First, in comparison to the vanilla dual encoder models such as Sentence-BERT, the training cost of VIRT is higher due to its introduction of virtual interaction distillation computation (i.e., the computational cost of distillation loss). Second, the performance of VIRT is highly correlated with the performance of the interaction-based teacher. Stronger teacher usually leads to the dual encoder student with higher performance.

Acknowledgements

This work was supported by the grants from National Natural Science Foundation of China (No.U20A20229, 62072423), CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-007B), and the USTC Research Funds of the Double First-Class Initiative (No.YD2150002009).

References

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-Yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6008–6019. Association for Computational Linguistics.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. Deformer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4487–4497. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174. Association for Computational Linguistics.

Jiecao Chen, Liu Yang, Karthik Raman, Michael Bendersky, Jung-Jung Yeh, Yun Zhou, Marc Najork, Danyang Cai, and Ehsan Emadzadeh. 2020. Dipair: Fast and accurate distillation for trillion-scaletext matching and pair modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2925–2937. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 670–680. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Deepak Gupta, Rajkumar Pujari, Asif Ekbal, Pushpak Bhattacharyya, Anutosh Maitra, Tom Geo Jain, and Shubhashis Sengupta. 2018. Can taxonomy help? improving semantic question matching using question taxonomy. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 499–513. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2042–2050.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on*

research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pages 39–48. ACM.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. In *Advances in Neural Information Processing Systems*, volume 34, pages 9204–9215.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1291–1299. ACM.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jianmo Ni, Gustavo Hernandez Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1864–1874. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Barun Patra. 2017. A survey of community question answering. *CoRR*, abs/1705.04009.

Joaquín Pérez-Iglesias, José R. Pérez-Agüera, Víctor Fresno, and Yuval Z. Feinstein. 2009. Integrating the probabilistic models BM25/BM25F into lucene. *CoRR*, abs/0911.5046.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *CoRR*, abs/1907.01041.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4322–4331. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2158–2170. Association for Computational Linguistics.

Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Improving document representations by generating pseudo query embeddings for dense retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*,

pages 5054–5064. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. *CoRR*, abs/2202.06991.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *CoRR*, abs/2002.10957.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3731–3741. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 496–505. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 287–296. ACM.

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 653–662. The Association for Computer Linguistics.