Tri-Attention: Explicit Context-Aware Attention Mechanism for Natural Language Processing

Rui Yu, Yifeng Li, *Member, IEEE,* Wenpeng Lu, *Member, IEEE,* and Longbing Cao, *Senior Member, IEEE*

Abstract—In natural language processing (NLP), the context of a word or sentence plays an essential role. Contextual information such as the semantic representation of a passage or historical dialogue forms an essential part of a conversation and a precise understanding of the present phrase or sentence. However, the standard attention mechanisms typically generate weights using query and key but ignore context, forming a Bi-Attention framework, despite their great success in modeling sequence alignment. This Bi-Attention mechanism does not explicitly model the interactions between the contexts, queries and keys of target sequences, missing important contextual information and resulting in poor attention performance. Accordingly, a novel and general triple-attention (Tri-Attention) framework expands the standard Bi-Attention mechanism and explicitly interacts query, key, and context by incorporating context as the third dimension in calculating relevance scores. Four variants of Tri-Attention are generated by expanding the two-dimensional vector-based additive, dot-product, scaled dot-product, and bilinear operations in Bi-Attention to the tensor operations for Tri-Attention. Extensive experiments on three NLP tasks demonstrate that Tri-Attention outperforms about 30 state-of-the-art non-attention, standard Bi-Attention, contextual Bi-Attention approaches and pretrained neural language models[1].

Index Terms—Attention mechanism, Context, Interaction, Triple attention, Natural language understanding

I. INTRODUCTION

ATTENTION is the cognitive process of selectively concentrating on one task while ignoring others. It efficiently allocates the finite brain processing resources to more important tasks and distributes the resources to prioritized tasks [1]. In neural networks, an attention mechanism is an adaptive sparse method for encoding and modelling sequence interactions by the association scores between different elements [2]. It has been widely applied in various tasks of natural language processing (NLP), including machine translation [3],

Rui Yu and Wenpeng Lu are with the Department of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China (e-mail: rui.yu1996@foxmail.com; wenpeng.lu@qlu.edu.cn).

Yifeng Li is with Department of Computer Science, Brock University, Niagara Region, ON L2S 3A1, Canada (e-mail: yli2@brocku.ca).

Longbing Cao is with the Data Science Institute, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: longbing.cao@uts.edu.au).

[1]The source codes and data are available at https://github.com/yurui12138/Tri-Attention.
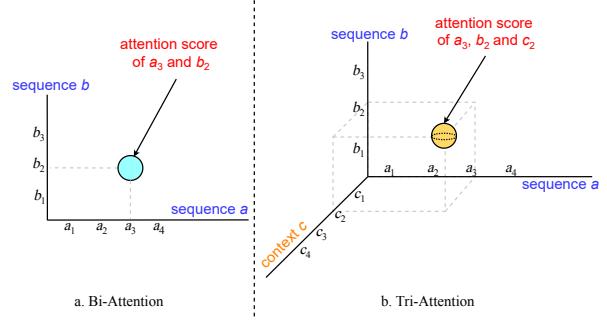

Fig. 1: Comparison of the standard Bi-Attention mechanism with our proposed Tri-Attention mechanism. a_i and b_j represent two words from two sequences respectively, and c_k denotes a contextual feature to them.

text matching [4], automatic question answering [5], and reading comprehension [6].

The attention mechanism learns an inner-representation of a sequence and the relationships between sequences, achieving great success in various tasks [7], [8], [9]. However, it does not explicitly involve context as human cognition works, where context plays a critical role in human visual and language comprehension. Human brain has neural networks dedicated to reading and providing context from the environment in order to improve perceptual learning and visual interpretation [10]. In natural language understanding, context is the key to derive and predict the meaning of a sentence [11]. It can improve language comprehensibility and how efficiently the comprehension is cognitively processed [12]. Different from this human attention mechanism, the existing attention mechanisms in deep neural networks usually concentrate on word-level feature interactions but fail to fully account for the overall context of the word or sentence [13]. Accordingly, the attention matrix is calculated just by two individual tokens *query* and *key* extracted from sequences without a full consideration of their context, forming a standard query-key-based two-dimensional attention (we call it Bi-Attention for simplicity) framework to capture sequence interactions, as shown in Fig. 1(a). This standard Bi-Attention framework requires a deeper understanding of the data to obtain the inner representation of sequences and capture the intrinsic relationships between sequences [14]. Consequently, without involving context, the existing attention models may not effectively capture really important context-aware information [13] and may produce unsatisfied to inaccurate attention. Some existing work has identified this design flaw in the Bi-Attention mechanism and

incorporated a contextual feature into the attention matrix. For example, the COIN and RE2 models align contextual features with attention to improve sentence representation learning, yielding better results than those without context [13], [15].

Although the importance of context has been recognized in the existing work, context has only been employed as extra supporting information to sequence representation learning. No general frameworks are available to simulate human attention with context and explicitly capture the interactions between target and contextual sequences or information. To manage this framework gap, we propose a triple-attention (Tri-Attention) framework to simulate human attention and explicitly capture interactions between sequences and between sequences and context. Different from existing Bi-Attention and contextual attention like COIN and RE2 models by integrating contextual features into queries and keys, our Tri-Attention mechanism directly involves context as the third dimension in quantifying sequence interactions. A query-key-context three-dimensional attention framework is formed for Tri-Attention, as shown in Fig. 1(b). The standard Bi-Attention mechanism adopts *query* and *key* to calculate two-dimensional relevance (e.g., dot product), forming an attention matrix. Then matrix operations are applied to the attention matrix and *value* to obtain the attention embedding. Differently, the attention matrix in Tri-Attention captures three-dimensional interactions between *query*, *key*, and *context*. To make *value* consistent with the semantic space of the triple attention matrix, the relevance between *context* and *value* is also calculated, producing *contextual value*. Finally, the attention embedding is obtained as a weighted linear combination of contextual value vectors. We test Tri-Attention-enabled networks for three independent NLP tasks, including retrieval-based English dialogue, Chinese sentence matching, and English reading comprehension. The results show that Tri-Attention outperforms most state-of-the-art approaches without attention, with Bi-Attention, contextual Bi-Attention, and pretrained language models with context. Furthermore, we evaluate the different performance of Bi-Attention, contextual Bi-Attention versus Tri-Attention and the effectiveness of Tri-Attention through a case study.

The main contributions of this work include:

- The proposed Tri-Attention mechanism expands the standard two-dimensional attention framework to explicitly involve and couple contextual information with query and key, hence the attention weights more sufficiently capture context-aware sequence interactions. To the best of our knowledge, this is the first work on explicitly involving context (contextual features) and learning query-key-context interaction-based attention between sequences.

- Tri-attention takes a general three-dimensional tensor framework, which can be instantiated into different implementations and applied to various tasks. We illustrate four variants by expanding the additive, dot-product, scaled dot-product and trilinear operations on query, key and context using tensor algebra for calculating Tri-Attention.

- Extensive experiments on three different NLP tasks and their real-world public datasets demonstrate the effectiveness of Tri-Attention. The Tri-Attention-enabled networks produce substantially better performance than

about 30 state-of-the-art methods.

The rest of this paper is organized as follows. Section II introduces the related work. Section III discusses the background and preliminaries. Section IV proposes Tri-Attention mechanism and its variant implementations. Section V introduces the multi-task-shared Tri-Attention network. Section VI demonstrates the performance of Tri-Attention by comparing it with state-of-the-art methods in terms of a variety of aspects. Lastly, Section VII concludes this work.

II. RELATED WORK

In this section, we briefly review the related work on attention mechanisms and involving contextual information in attention, respectively.

Human perception and visual processing may selective certain relevant parts of an image while ignoring other irrelevant information [1]. This human attention mechanism has inspired the neural attention models widely applied in computer vision [16]. Attention has also shown success in neural NLP tasks, such as machine translation, automatic question answering, sentence matching, and word sense disambiguation, where attention usually captures the interactions between sequences. For example, in machine translation, a recurrent attention measures the weights of all heads in each Transformer layer to build more efficient neural machine translation models [17]. In [18], syntax-enhanced attention mechanisms SEAs improve syntactic-enhanced machine translation by involving a dependency mask bias and a relative local-phrasal position bias. In [19], a source-target bilingual syntactic alignment SyntAligner aligns the syntactic structures of source and target sentences with border-sensitive span attention and then maximizes their mutual dependency for machine translation. In automatic question answering, a Block-Skim attention mechanism measures the importance of tokens and blocks to skim unnecessary context, improving the Transformer performance [20]. In sentence semantic matching (SSM), A 3D CNN-based SSM model first constructs multi-dimensional representations for each sentence, then utilizes a 3D CNN and an attention mechanism to learn the interactive matching features [21]. In word sense disambiguation (WSD), an extractive sense comprehension mechanism employs local self-attention to constraint attentions to be local, which allows to handle longer context without heavy computing burdens [22].

In NLP tasks, the existing Bi-Attention mechanisms calculate attention scores by focusing on local relevance matching at the token level without explicitly taking their overall context into account. Each element of the attention matrix is only computed based on two individual tokens *query* and *key*. This two-dimensional Bi-Attention mechanism ignores the interactions with and influence of their context, e.g., contextual words or sentences surrounding target words. On one hand, contextual information may involve important information and influence on the target, as demonstrated in other areas such as contextual and sequential recommender systems [23] and contextual matrix factorization [24]. On the other, the interactions and couplings between a target and its context form essential constituents of the target, as shown in attribute/feature

interaction analysis [25], [26], and multi-party interaction learning [27], [28]. These prompt the potential and need for including context and target-context interactions into neural attention mechanisms.

In fact, various efforts aim to involve contextual information in Bi-Attention mechanisms for NLP [13], [29], [30]. For example, Yang et al. [29] directly incorporate contextual information into the representations of query and key with addition operations. Ding et al. [30] argue that the existing cross-attention was confused by the localness perception problem, which fails to adequately capture the whole context. A context-aware cross-attention method models both local and global contexts, which uses an interpolation gating mechanism to combine the original and local cross-attention. Hu et al. [13] propose a context-aware attention network (COIN) for sentence matching. Its core component is a context-aware interaction block consisting of a context-aware cross-attention layer and a gate fusion layer, which consults contextual information to enable better alignments and blends the original and aligned representations with a gate connection.

In addition, contextual information or contextual attention has also been applied in various non-NLP tasks, such as image inpainting [31], [32], video captioning [33], recommendation [23], [34], and decision making [35], [36]. Regarding contextual attention for various vision and image-related tasks, for instance, in [31], contextual attention in CNN synthesizes both image structures and surrounding image features as references for image inpainting. The selective contextual attention in [37] further learns pixel- and patch-level attentions from background regions and selectively utilizes contextual attention to enhance the original features. The contextual attention network in [33] first extracts visual and textual features at each time step, then utilizes a contextual attention mechanism to capture more information for captioning. In contextual attention-based recommendation, for example, attention-based influential contexts are modeled by involving heterogeneous relations for recommendation [34]. In [23], the attention of context is measured and incorporated into sequential recommendation. These studies show the great potential of integrating context with attention for non-NLP tasks. Furthermore, in contextual reinforcement learning (cRL), a contextual MDP (cMDP) extended the standard Markov Decision Process (MDP) to learn context-conditioned policies, which increases the robustness and generalization of RL models [35], [36].

Although the above-mentioned work strives to implement contextual attention to model the influence from context, they do not directly capture target-context interactions, i.e., the interactions between *context*, *query* and *key* in calculating attention scores. They still rely on the framework of standard query-key-based Bi-Attention mechanism, merely integrate contextual information and the original representation using addition operations or gating mechanisms, there are no-to-weak interactions between contextual information and the representations of *query* and *key*.

To address the shortage of two-dimensional query-key attention and the limitation of existing studies, we propose a query-key-context triple attention (Tri-Attention) framework. Tri-Attention uses tensor algebra techniques to explicitly in-volve contextual information and capture query-key-context interactions. In calculating the attention matrix, Tri-Attention incorporates contextual information and treats it equally with the Bi-Attention factors *query*, *key*, and *value*. In this way, Tri-Attention expands Bi-Attention mechanisms by allowing contextual information to explicitly participate in calculating attention weights.

Consequently, the contextual information in Tri-Attention plays an essential role in capturing the interactions between sequences under their contexts. This also expands the contextual Bi-Attention mechanisms where context only plays a complementary role.

III. BACKGROUND AND PRELIMINARIES

Here, we introduce tensor algebra and the design of the standard Bi-Attention mechanism. These lay the foundation of our proposed Tri-Attention.

A. Tensor Algebra

Our Tri-Attention builds on tensor algebra to incorporate and interact context with query, key, and value and calculate attention weights on the query-key-context tensor. Therefore, here, we introduce the essential algebraic notations and concepts. We use a bold uppercase symbol to represent a matrix, e.g., $\boldsymbol{X}$; a bold lowercase for a vector (column), e.g., $\boldsymbol{x}$; and a lowercase or uppercase for a scalar, e.g., x or X. A tensor is denoted by a bold calligraphic symbol, e.g., a three-dimensional real-valued tensor: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$.

The tensor or vector matrix multiplication [38] is applied to calculate Tri-Attention weights. Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and a matrix $\boldsymbol{Y} \in \mathbb{R}^{J \times I_n}$, the n-mode product of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{Y}$ is defined as $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{Y} = \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$, where

$$\boldsymbol{\mathcal{Z}}_{i_1 i_2 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} y_{j i_n}. \quad (1)$$

Similarly, the n-mode product between $\boldsymbol{\mathcal{X}}$ and column vector $\boldsymbol{y} \in \mathbb{R}^{I_n}$ is written as $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{y}^\mathsf{T} = \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times 1 \times I_{n+1} \times \cdots \times I_N}$ with element-wise entry as:

$$\boldsymbol{\mathcal{Z}}_{i_1 i_2 \cdots i_{n-1} 1 i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_n \cdots i_N} y_{i_n}. \quad (2)$$

Oftentimes, the trivial n-th dimension in $\boldsymbol{\mathcal{Z}}$ is squeezed out such that $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$ is $(n-1)$-way. In this work, we keep this dimension for the convenience in formulating the Trilinear attention in Section IV.

To formulate our idea, we use $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_J] \in \mathbb{R}^{D \times J}$ to represent the contextual matrix which contains J context vectors. Each contextual vector is a column vector of length D, that is $\boldsymbol{c}_j \in \mathbb{R}^D$, $j = 1, 2, \cdots, J$. In this paper, a vector refers to a column vector by default. The key information is represented by matrix $\boldsymbol{K} = [\boldsymbol{k}_1, \boldsymbol{k}_2, \cdots, \boldsymbol{k}_I] \in \mathbb{R}^{D \times I}$ which includes I key vectors of D dimensions. Similarly, the corresponding value information is represented by $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_I] \in \mathbb{R}^{D \times I}$. In general, a query vector is denoted by $\boldsymbol{q} \in \mathbb{R}^D$. For N query vectors, we use $\boldsymbol{Q} \in \mathbb{R}^{D \times N}$.

B. Standard Bi-Attention Mechanism

The standard Bi-Attention mechanism customarily obtains interactive features by calculating the relevance score between sequences. It consists of three major steps.

First, the relevance $F(\boldsymbol{q}, \boldsymbol{k}_i)$ between query $\boldsymbol{q}$ and key $\boldsymbol{k}_i$ is calculated:

$$F(\boldsymbol{q}, \boldsymbol{k}_i) = \text{Similarity}(\boldsymbol{q}, \boldsymbol{k}_i), \quad i = 1, 2, \cdots, I, \quad (3)$$

Second, $F(\boldsymbol{q}, \boldsymbol{k}_i)$ is normalized by the softmax function:

$$\alpha_i = \frac{\exp\left(F(\boldsymbol{q}, \boldsymbol{k}_i)\right)}{\sum_{i'=1}^{I} \exp\left(F(\boldsymbol{q}, \boldsymbol{k}_{i'})\right)}, \quad i = 1, 2, \cdots, I, \quad (4)$$

where α_i is the normalized attention weight.

Lastly, the attention embedding is obtained by a weighted linear combination with value $\boldsymbol{v}_i$:

$$\boldsymbol{q}_{\text{new}} = \sum_{i=1}^{I} \alpha_i \boldsymbol{v}_i = \boldsymbol{V}\boldsymbol{\alpha}. \quad (5)$$

In practice, the similarity measure is not unique, resulting in different attention mechanisms and designs. The four most commonly used similarity measure methods are additive (Add) similarity, dot-product (DP) similarity, scaled dot-product (SDP) similarity, and bilinear (Bili) similarity.

Additive (Add) similarity [39]:

$$F(\boldsymbol{q}, \boldsymbol{k}_i) = \boldsymbol{p}^{\mathsf{T}}\tanh(\boldsymbol{W}\boldsymbol{q} + \boldsymbol{U}\boldsymbol{k}_i), \quad i = 1, 2, \cdots, I, \quad (6)$$

Dot-product (DP) similarity [40]:

$$F(\boldsymbol{q}, \boldsymbol{k}_i) = \boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_i, \quad i = 1, 2, \cdots, I, \quad (7)$$

Scaled dot-product (SDP) similarity [41]:

$$F(\boldsymbol{q}, \boldsymbol{k}_i) = \frac{\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_i}{\sqrt{D}}, \quad i = 1, 2, \cdots, I, \quad (8)$$

Bilinear (Bili) similarity [40]:

$$F(\boldsymbol{q}, \boldsymbol{k}_i) = \boldsymbol{q}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{k}_i, \quad i = 1, 2, \cdots, I, \quad (9)$$

$\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{p}$ are learnable parameters.

The main drawback of the standard Bi-Attention mechanisms lies in that the relevance scores between sequences only build on the token representations of $\boldsymbol{q}$ and $\boldsymbol{k}_i$. No contextual information and the interaction between context and query and key are involved, which may miss important environmental information and cause inaccurate relevance.

IV. TRI-ATTENTION MECHANISM

In this section, we introduce the proposed Tri-Attention mechanism. It extends the Bi-Attention mechanisms and builds on tensor operations for variant implementations of the query-key-context similarity.
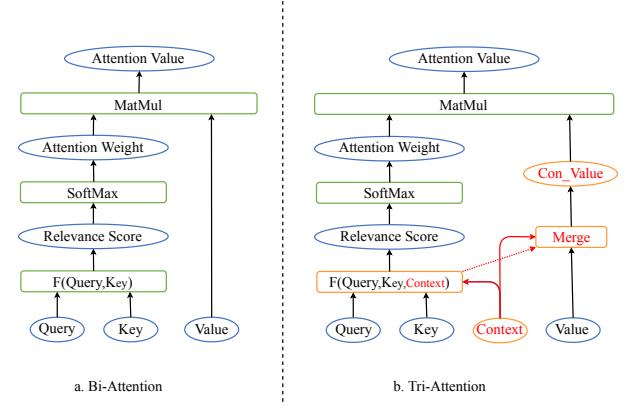

Fig. 2: Comparison of the standard Bi-Attention versus Tri-Attention mechanisms. The elements in red refer to the new modules introduced by Tri-Attention, which adjust standard query-key-value to be context dependent. As a result, the calculation of contextual relevance scores is semantically consistent with the computation of contextual values.

A. Framework of Tri-Attention Mechanism

The Tri-Attention mechanism expands the standard query-key similarity-based Bi-Attention mechanism in Section III-B to query-key-context tensor similarity, which thus explicitly involves contextual information and captures the interactions between target sequences and their contexts. Tri-Attention engages contextual features in calculating the relevance scores between sequences and then adjusts values to be context dependent. Bi-Attention can be viewed as a special case of Tri-Attention without the explicit dimension of context. Once *context* is removed, Tri-Attention is degraded to Bi-Attention. Fig. 2 illustrates the main ideas and processes of Bi-Attention versus Tri-Attention mechanism. The extension of Bi-Attention to Tri-Attention is marked in red for differentiating the two mechanisms.

In contrast to the standard three-step attention learning in Bi-Attention, Tri-Attention generalizes their first and third steps. In the first step, Tri-Attention first expands the similarity calculation by involving contextual information to obtain context-dependent query-key relevance scores, forming *contextual relevance score*. In the third step, Tri-Attention explicitly integrates context and value to produce context-dependent value, resulting in *contextual value*. The resultant contextual value resides in the same semantic space as contextual relevance scores. Both capture extra contextual information and query-key-value-context interactions for a more informative but semantically consistent attention representation.

B. Contextual Relevance Score

To obtain relevance scores informative to the context of target sequences, we expand the relevance calculation methods including additive, dot-product, scaled dot-product, and bilinear similarity in Section III-B by explicitly incorporating contextual information as a third dimension. This results in four contextual relevance similarity calculators: T-additive (TAdd), T-dot-product (TDP), T-scaled-dot-product (TSDP),

and Trilinear (Trili) operations on the query-key-context tensor by applying the tensor-matrix or vector product operations defined in Eqs. (1) and (2). Here, prefix "T-" indicates its roles for Tri-Attention by involving the third dimension context into attention mechanisms.

T-additive (TAdd) similarity: The T-additive similarity expands the additive similarity in Eq. (6) by adding a context dimension c_j to the usual terms q and k_i, formulated below:

$$F(q, k_i, c_j) = p^\mathsf{T}\tanh(Wq + Uk_i + Hc_j),$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J \qquad (10)$$

where W, U, H, and p are learnable parameters.

T-dot-product (TDP) similarity: The T-dot-product similarity expands the standard query-key dot-product attention in Eq. (7) for Bi-Attention to the query-value-context T-dot-product attention. T-dot-product attention replaces the inner product of query and value vectors by the *contextual inner product* between the query, value, and context vectors, which is formulated below:

$$F(q, k_i, c_j) = \sum_{d=1}^{D} q_d k_{id} c_{jd} = \langle q, k_i, c_j \rangle$$
$$= \mathcal{I} \times_1 q^\mathsf{T} \times_2 k_i^\mathsf{T} \times_3 c_j^\mathsf{T} \qquad , \qquad (11)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J$$

where $\langle q, k_i, c_j \rangle$ is the contextual inner product of three vectors; q^T, k_i^T, and c_j^T, which are treated as row vectors, i.e. of size $1 \times D$. $\times_1$, $\times_2$, and $\times_3$ are 1-mode, 2-mode, and 3-mode tensor-matrix or vector multiplication operators, respectively. $\mathcal{I}$ is a 3-way identity tensor of size $D \times D \times D$, where only the (d, d, d)-th element is 1 ($d = 1, 2, \cdots, D$), and others are 0. The resultant $F(q, k_i, c_j)$ is a scalar (i.e., of size $1 \times 1 \times 1$).

T-scaled-dot-product (TSDP) similarity: Similarly, the scaled-dot-product attention in Eq. (8) for Bi-Attention is generalized to T-scaled-dot-product. T-scaled-dot-product divides the contextual inner product by the squared root of the number of dimensions:

$$F(q, k_i, c_j) = \frac{\sum_{d=1}^{D} q_d k_{id} c_{jd}}{\sqrt{D}} = \frac{\langle q, k_i, c_j \rangle}{\sqrt{D}}$$
$$= \frac{\mathcal{I} \times_1 q^\mathsf{T} \times_2 k_i^\mathsf{T} \times_3 c_j^\mathsf{T}}{\sqrt{D}} \qquad . \qquad (12)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J$$

Trilinear (Trili) similarity: With context involved, the bilinear form in Eq. (9) is naturally extended to a multilinear, precisely trilinear, form using tensor-vector products, labeled as Trilinear:

$$F(q, k_i, c_j) = \sum_{d=1}^{D} \sum_{d'=1}^{D} \sum_{d''=1}^{D} w_{dd'd''} q_d k_{id'} c_{jd''}$$
$$= \mathcal{W} \times_1 q^\mathsf{T} \times_2 k_i^\mathsf{T} \times_3 c_j^\mathsf{T} \qquad , \qquad (13)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J$$

The learnable weight tensor $\mathcal{W} \in \mathbb{R}^{D \times D \times D}$ governs the interactions between any dimensions of the three vectors. T-dot-product and T-scaled-dot-product are special cases of this

general form. A nice property of this contextual generalization is that the contextual relevance scores can be computed and stored in a tensor using query, key, and context matrices: $\mathcal{F}(Q, K, C) = \mathcal{W} \times_1 Q^\mathsf{T} \times_2 K^\mathsf{T} \times_3 C^\mathsf{T} \in \mathbb{R}^{N \times I \times J}$ where N, I, and J are the number of query vectors, key vectors, and context vectors, respectively. However, the size of $\mathcal{W}$ grows in a cubic way. Hence, $\mathcal{F}(Q, K, C)$ may not scale well. In practice, we can use an economic version as follows:

$$F(q, k_i, c_j) = \langle Wq, Uk_i, Hc_j \rangle$$
$$= \mathcal{I} \times_1 (Wq)^\mathsf{T} \times_2 (Uk_i)^\mathsf{T} \times_3 (Hc_j)^\mathsf{T}, \qquad (14)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J$$

where W, U, and H are learnable matrices. Concisely, for matrix inputs, we have $\mathcal{F}(Q, K, C) = \mathcal{I} \times_1 (WQ)^\mathsf{T} \times_2 (UK)^\mathsf{T} \times_3 (HC)^\mathsf{T} \in \mathbb{R}^{N \times I \times J}$.

C. Normalized Contextual Relevance Score

Consistent with the Bi-Attention mechanism, the contextual relevance score $F(q, k_i, c_j)$ calculated by Eq. (14) for Tri-Attention is also normalized by the softmax function:

$$\alpha_{ij}^c = \frac{\exp\left(F(q, k_i, c_j)\right)}{\sum_{i'=1}^{I} \sum_{j'=1}^{J} \exp\left(F(q, k_{i'}, c_{j'})\right)}. \qquad (15)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J$$

Consequently, the attention weights for query q is represented by matrix $A^c \in \mathbb{R}^{I \times J}$.

D. Contextual Value

The above contextual relevance scores contain contextual information and interactions between context, query, and key. To semantically match value to this contextual relevance score, we further integrate value with context using one of the following methods to obtain contextual value.

Additive context-value integration:

$$v_{(i,j)}^c = v_i + c_j \quad i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J. \qquad (16)$$

Multiplicative context-value integration:

$$v_{(i,j)}^c = v_i * c_j \quad i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J \qquad (17)$$

where $*$ is the Hadamard product operator.

Bilinear context-value integration:

$$v_{(i,j)}^c = (U'v_i) * (H'c_j)$$
$$i = 1, 2, \cdots, I; \; j = 1, 2, \cdots, J. \qquad (18)$$

In Eqs. (16)-(18), $v_{(i,j)}^c \in \mathbb{R}^D$. These methods generate the contextual value tensor $\mathcal{V}^c \in \mathbb{R}^{I \times J \times D}$.

E. Contextual Attention Embedding

Since there are different approaches to obtain contextual attention weight matrix A^c by Eq. (15) and value tensor $\mathcal{V}^c$ per Eqs. (16)-(18), for the semantic consistency, their operation may be consistent. For example, if T-additive in Eq. (10) is applied to obtain α_{ij}^c, then additive context-value integration in Eq. (16) should be correspondingly applied to
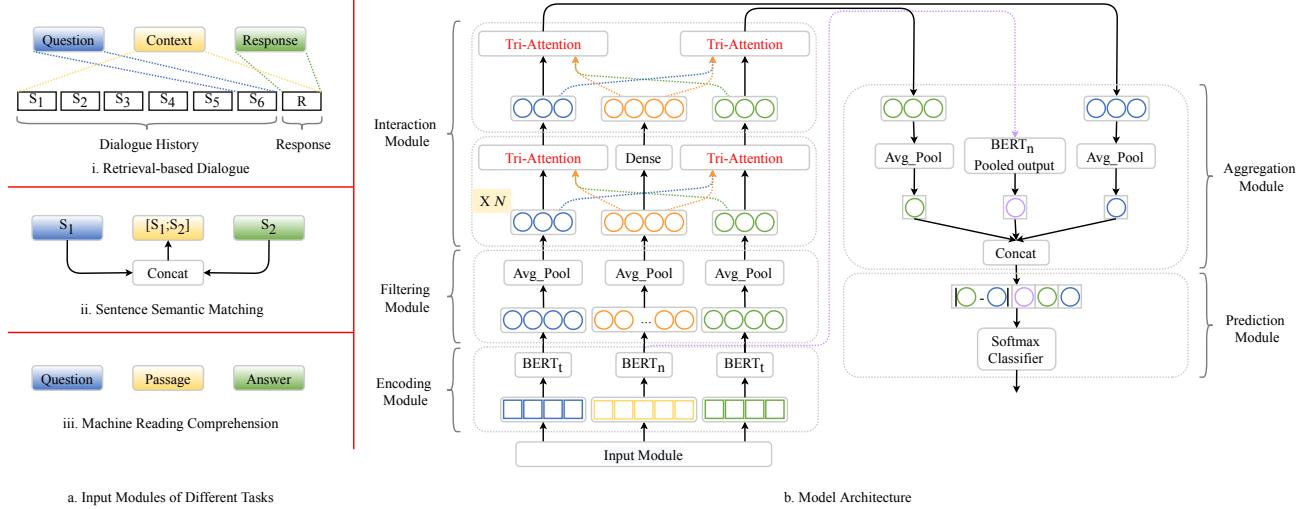
Fig. 3: Shared architecture of Tri-Attention network (TAN) for three NLP tasks: left panel - input modules, right panel - model architecture.

obtain contextual value $\boldsymbol{v}^c_{(i,j)}$. Further, the new *contextual attention embedding* $\boldsymbol{q}^c_{\text{new}}$ corresponding to query $\boldsymbol{q}$ is:

$$\boldsymbol{q}^c_{\text{new}} = \sum_{i=1}^I \sum_{j=1}^J \alpha^c_{ij} \boldsymbol{v}^c_{(i,j)} = \boldsymbol{V}^c \boldsymbol{\alpha}^c, \qquad (19)$$

where $\boldsymbol{q}^c_{\text{new}} \in \mathbb{R}^D$, $\boldsymbol{V}^c$ is mode-3 matricized from $\boldsymbol{\mathcal{V}}^c$: $\boldsymbol{V}^c = [\boldsymbol{v}^c_1, \boldsymbol{v}^c_2, \cdots, \boldsymbol{v}^c_{I*J}] \in \mathcal{R}^{D \times (I*J)}$, and $\boldsymbol{\alpha}^c$ is vectorized from $\boldsymbol{A}^c$. For $m = (i-1)I + j$, $\boldsymbol{v}^c_m$ in $\boldsymbol{V}^c$ corresponds to $\boldsymbol{v}^c_{(i,j)}$ in $\boldsymbol{\mathcal{V}}^c$ and α^c_m in $\boldsymbol{\alpha}^c$ is the same as α^c_{ij} in $\boldsymbol{A}^c$.

F. Choice of Contexts

Since contextual information varies over tasks and definitions, the acquisition of context is task-specific [13], [15], [30], [36], [42]. As an example by following the practice in [42] for NLP tasks, we first choose the encoding results of BERT for different sequences as the preliminary contextual information. Then an average pooling operation is applied to obtain the final contextual features.

More specifically, assume a task consists of several input sequences S_1, S_2, $\cdots$, and S_K. First, we segment each sequence according to the minimum granularity of the language underlying the inputs. For example, if the sequences are in Chinese, Chinese can be segmented in terms of Chinese characters. In contrast, if English is involved, the inputs should be segmented on the granularity of words. Then, we concatenate the segmented sequences with segments $\{SEP\}$ to a new sequence as follows:

$$S = \{[CLS], S_1, [SEP], \cdots, [SEP], S_K, [SEP]\}$$
$$= \{[CLS], w_1^1, \cdots, w_1^{l_{S_1}}, [SEP], \cdots, [SEP], \qquad (20)$$
$$w_K^1, \cdots, w_K^{l_{S_K}}, [SEP]\},$$

where w_k^l is the l-th token in the k-th sequence. Finally, we feed this new sequence S into BERT to obtain the contextual information, which consists of a list of vectors corresponding to all tokens respectively. It is worth noting that the output of

BERT consists of two parts, and we use the first part as the contextual feature.

V. A MULTI-TASK-SHARED TRI-ATTENTION NETWORK

Here, we apply Tri-Attention in diverse NLP tasks by constructing a multi-task-shared Tri-Attention-enabled network (TAN). We test Tri-Attention for retrieval-based dialogue, sentence semantic matching, and machine reading comprehension. Accordingly, the shared architecture of our Tri-Attention network for these tasks is shown in Fig. 3.

TAN consists of two main panels. The left consists of three input modules corresponding to three NLP tasks, respectively. For each task, its input sequences and contexts specified according to the task-specific requirement, which are marked by different colors. The right panel implements the corresponding attention mechanism and learning tasks. It consists of five core modules: encoding module, filtering module, interaction module, aggregation module, and prediction module. First, the encoding module employs BERT$_t$ to encode the two input sequences to obtain their feature representations, and utilizes BERT$_n$ to encode the context to obtain the contextual feature representation. Second, the filtering module trims the sequence length of the above representations to facilitate the following processing. Third, the interaction module applies Tri-Attention to engage three representations of input and context sequences and then couple them by the Tri-Attention mechanism. The stack number (i.e., N in Fig. 3) of interaction modules is adjustable per learning task requirements and based on the relevance score calculation methods, as shown in Table IV. Furthermore, the feature aggregation module performs the average-pooling of the representations of two input sequences to filter them and then concatenates their representations with the pooling representation of the output of BERT$_n$, and the difference between representations of two input sequences. Lastly, a prediction module feeds the above four representations to a softmax function to predict the final response matching score. Below, in discussing the experimental settings

for each task, we will further explain the task-specific settings to customize TAN for each task.

VI. Experiments

To verify the effectiveness of Tri-Attention mechanism in comparison with Bi-Attention, we conduct extensive experiments on three NLP tasks: retrieval-based dialogue, sentence semantic matching, and machine reading comprehension. Three public datasets are used to evaluate the results by comparing with different baseline methods for each task. In addition, we report the results of ablation study and case study.

A. Evaluation Tasks and Their Datasets

We evaluate Tri-Attention in three diverse NLP tasks: retrieval-based dialogue, sentence semantic matching, and machine reading comprehension. These tasks and their corresponding datasets are introduced below.

- *Retrieval-based dialogue*. Given historical dialogue utterances, this task selects the correct response from multiple candidate responses. A commonly used data *Ubuntu Corpus V1* [43] is tested here. It is a public dialogue dataset containing some 1 million multi-turn dialogues, with a total of over 7 million utterances and 100 million words. The ratio of positive versus negative instances in the training set is 1:1. The ratio of positive versus negative instances in the validation and test sets is 1:9. Table I illustrates the samples from the Ubuntu Corpus V1 corpus. S_1-S_6 are historical dialogue utterances, and the candidate response utterances consist of a positive case and a negative case.

Historical dialogue utterances	S_1	hey guys i am trying to compile an application __path__ went well and it created a __path__ which takes some arguments but i dont know how to use this file ... can somebody give me a hint
	S_2	did n't __path__ created a makefile
	S_3	ikonia take a look at this __url__
	S_4	it did but is does nothing
	S_5	it just mentions make *** no targets specified and no makefile found stop ."
	S_6	seems like __path__ uses this .. you do n't have to run this manually
Candidate response utterances	Positive (label:1)	look at the install file i think i just have to pass the right arguments
	Negative (label:0)	thanks a lot i think i have more to go on for my little project the syntax there there is pretty easy to parse and all you need is to make wget download that url

TABLE I: Instances of retrieval-based dialogues [43].

- *Sentence semantic matching*. This task decides whether two sentences share similar meaning. We use a large-scale Chinese corpus LCQMC [44] for sentence semantic matching. It consists of 260,068 question pairs collected by Baidu Knows. There are three subsets: 238,766 question pairs for training, 8,802 question pairs for validation, and 12,500 question pairs for test. Table II illustrates two of its instances in LCQMC.

Positive (label:1)	S_1	哪首歌里有这句歌词 (En: Which song has this lyrics)
	S_2	这句歌词是哪首歌的? (En: Which song does this lyric come from?)
Negative (label:0)	S_3	哪些浏览器可以看电影 (En: Which browser can play movies)
	S_4	什么浏览器可以下电影 (En: Which browser can download movies)

TABLE II: Instances of sentence semantic matching data [44].

- *Machine reading comprehension*. Given a passage and its corresponding question, this task identifies the correct answer from multiple candidate choices. RACE [45] is recognized as one of the largest and most difficult English datasets for multi-choice reading comprehension. It consists of two subsets: RACE-M and RACE-H, corresponding to the difficulty level for middle school and high school, respectively. Table III illustrates this data.

Passage	Are you carrying too much on your back at school? You're not alone. Back experts in the USA are worried about that young students are having back and neck problems because they are carrying too much in their backpacks ... (3) The heaviest things should be packed closest to the back. (4) Bend both knees when you pick up the pack, don't just bend over the waist
Question	The main idea of the passage is about __.
Candidate choices	A. the problems made by rolling backpacks B. the advantage of backpacks C. the best backpacks for students D. how to lighten students' backpacks
Correct answer	D

TABLE III: Instances of machine reading comprehension data [45]

B. TAN Settings

As shown in Fig. 3, the contextual information in Tri-Attention the concatenation of the outputs of BERT applied on input sequences, consistent with [42]. It is shown in [46] that the outputs of different layers of BERT can capture different sentence features. Inspired by this, our experiment shows that better performance can be achieved using the output of layer 1 as the representation of a single sequence. When the final representations of different sequences are obtained, we filter the representations by average pooling. Finally, inspired by [47], we concatenate the representations of different sequences, the differential representations between sequences, and the pooled representation of BERT. The concatenate results are fed to softmax for classification. We also add the dropout strategy with dropout rate 0.1 following [48]. The above settings are applied to all three tasks: retrieval-based dialogue, sentence semantic matching, and machine reading comprehension.

In addition, some hyperparameters are set for different tasks. The learning rate 1e-5 is used for retrieval-based dialogue, and

Task	Dataset	Tri-Attention$_{TAdd}$	Tri-Attention$_{TDP}$	Tri-Attention$_{TSDP}$	Tri-Attention$_{Trili}$
Retrieval-based dialogue	Ubuntu Corpus V1	3	2	3	4
Sentence semantic matching	LCQMC	2	3	3	1
Machine reading comprehension	RACE	4	3	4	1

TABLE IV: Number of Tri-Attention layers with different similarity operations in TAN for different NLP tasks and datasets.

2e-5 for both sentence semantic matching and machine reading comprehension. The batch size for retrieval-based dialogue is 32, with 64 for sentence semantic matching and 16 for machine reading comprehension. The cross-entropy loss is used for the objective function of all tasks. AdamW [49] is used for the optimization of retrieval-based dialogue and sentence semantic matching, and BertAdam [50] for machine reading comprehension. In TAN, Tri-Attention is stackable, thus the number of its layers is adjusted for different tasks. Table IV describes the number of layers for different tasks and datasets under different Tri-Attention similarity operations.

C. Task 1: Retrieval-based Dialogue

Here, we evaluate TAN with Tri-Attention mechanism against three categories of baselines for retrieval-based dialogue. The baselines consist of non-attention classic models, standard Bi-Attention-based neural networks with context or interaction, and pretrained neural language networks.

1) Baseline Methods: First, several non-attention classic methods are compared with TAN:

- *TF-IDF*: A statistical method obtains the relevance features of sentences per the frequency of characters [43].
- *RNN*: A simple RNN encodes utterances to obtain a feature representation and calculates the relevance score based on the representation [43].
- *CNN*: A CNN extracts utterance features and calculates the relevance representation of the features [51].
- *LSTM*: LSTM serves as the context encoder to extract utterance features and then calculates the relevance representation [51].

Second, several advanced neural networks with standard Bi-Attention mechanisms with context or interaction are compared with TAN:

- *SMN*: A sequential matching network selects responses to multi-turn conversations, which matches a response to each utterance in the context at multi-level granularities and utilizes RNN to accumulate the matching vectors to model the relations between utterances [52].
- *DUA*: A self-matching attention routes the vital information in each utterance then matches a response to each refined utterance, followed by attention mechanism to aggregate matching vectors to obtain the final matching score [53].
- *DAM*: It constructs the representations of utterances at different granularities solely with stacked self-attention then calculates the matching score between the context and response by cross-attention [54].
- *IoI*: An interaction-over-interaction network performs matching by stacking multiple interaction blocks to

achieve deep interaction between responses and utterances [55].
- *ESIM*: A sequential matching model based only on chain sequence to select multi-turn responses, which involves an enhanced sequential inference model and soft-alignment attention [56].
- *MSN*: A multi-hop selector network selects the relevant utterances as the context then calculates the matching score between the context and response [57].

Lastly, we compare TAN with several pretrained neural language models:

- *BERT*: A classical pretrained language model, which is fine-tuned for retrieval-based dialogue [50].
- *RoBERTa-SS-DA*: A RoBERTa-based approach, where a speaker segmentation scheme and dialogue augmentation improve the performance [58].
- *BERT-DPT*: It enhances the BERT ability by post-training on domain-specific corpus to train contextualized representations that are missing in general corpus [59].
- *BERT-VFT*: A fine-tuned model based on BERT, which involves parameter-efficient transfer learning for various tasks [60].
- *SA-BERT*: A speaker-aware BERT-based model, which perceives the change of speaker information and incorporates domain knowledge into pretrained BERT [61].
- *UMS$_{BERT+}$*: Utterance manipulation strategies are used to select multi-turn responses, including utterance insertion, deletion, and search strategies, to address some deficiencies in existing BERT models [62].
- *BERT-SL*: A context-response matching model based on pretrained language models, which introduces four self-supervised tasks to jointly train the response selection model in a multi-task manner [63].
- *BERT-UMS+FGC*: A fine-grained contrastive learning method for response selection, which generates better matching representation at finer granularity to select positive responses [64].

2) TAN Settings: The TAN with Tri-Attention in Fig. 3 is specified as follows for retrieval-based dialogue. The inputs shown on the left panel consist of question, response, and context. A question is the last sentence in the dialogue history and a response is the original response in the dataset. The context is the concatenation of dialogue and response [42]. The model on the right panel has five core modules, as described in Section V. The performance is evaluated by R$_n$@k, which denotes whether the top-k retrieved responses from the n candidate responses contain the right responses.

3) Performance Evaluation: The performance of TAN against all baseline models is shown in Table V. We obtain the following observations.

First, compared with the non-attention classic methods, the advantage of TAN is very significant. The classic methods model sentences only using observable word features and simple interaction patterns to evaluate each candidate response. They are unable to effectively utilize latent semantic information and learn complex interaction patterns. Thanks to the powerful pretrained language model and the Tri-Attention mechanism, TAN learns more sophisticated semantic features than these baselines. The learned contextual features are beneficial for representing utterances and responses, thus measuring their relevance more accurately.

Second, compared with Bi-Attention and interaction-based networks, TAN also performs much better. This is probably because TAN employs the pretrained $BERT_{base}$ model trained on large corpus, which enhances our model with great prior knowledge. In contrast, the benchmarks do not involve the vast amount of prior knowledge.

Third, TAN also outperforms the pretrained BERT, which is the state-of-the-art language model with self-attention mechanism. Although BERT gains rich prior knowledge pretrained on a vast amount of textual data, it may not capture interactions and contexts between utterances and responses for this task. In addition, BERT pretrained on a general corpus may not be adaptive to this domain-specific dialogue dataset. In contrast, TAN applies Tri-Attention to explicitly capture contexts and adopts the post-training strategy in [59] on the task dataset to optimize its specificity and adaptability.

Lastly, TAN with Tri-Attention consistently outperforms all pretrained BERT variants. Although most baselines also perform post-training, they may fail to incorporate contextual information and accurately capture interactive features between utterances and responses. In contrast, Tri-Attention fully considers the context in its attention mechanism to determine the relevance between utterances and responses, contributing to the improved performance of TAN over the BERT variants.

D. Task 2: Sentence Semantic Matching

We further test TAN with Tri-Attention mechanism against three categories of baselines for sentence semantic matching. The baselines consist of non-attention classic networks, more advanced networks with relations or standard Bi-Attention-based context, and pretrained neural language networks.

1) Baseline Methods: First, several non-attention classic networks are compared to TAN with Tri-Attention:

- *WMD_{char}* and *WMD_{word}*: They apply the Wasserstein distance to the matching degree between two sentences in terms of character and word, respectively [44].
- *CNN_{char}* and *CNN_{word}*: They utilize a CNN to encode two sentences for their corresponding sentence representations, which are then concatenated to predict the matching degree with a softmax classifier [44].
- *$BiLSTM_{char}$* and *$BiLSTM_{word}$*: They are similar with the former CNN method replacing CNN by a BiLSTM component [44].

Second, more advanced networks with relations or standard Bi-Attention-based context are compared with TAN embedded with Tri-Attention mechanism:

Methods	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
TF-IDF	41.0	54.5	70.8
RNN	40.3	54.7	81.9
CNN	54.9	68.4	89.6
LSTM	63.8	78.4	94.9
SMN	72.6	84.7	96.1
DUA	75.2	86.8	96.2
DAM	76.7	87.4	96.9
IoI	79.6	89.4	97.4
ESIM	79.6	89.4	97.5
MSN	80.0	89.9	97.8
BERT	80.8	89.7	97.5
RoBERTa-SS-DA	82.6	90.9	97.8
BERT-DPT	85.1	92.4	98.4
BERT-VFT	85.5	92.8	98.5
SA-BERT	85.5	92.8	98.3
UMS_{BERT+}	87.5	94.2	98.8
BERT-SL	88.4	94.6	99.0
BERT-UMS+FGC	**88.6**	**94.8**	**99.0**
TAN_{TAdd}	90.5 ($\pm$0.3)	95.8 ($\pm$0.06)	99.2 ($\pm$0.05)
TAN_{TDP}	90.3 ($\pm$0.1)	**95.9** ($\pm$0.3)	**99.3** ($\pm$0.06)
TAN_{TSDP}	**90.6** ($\pm$0.3)	95.7 ($\pm$0.05)	99.2 ($\pm$0.04)
TAN_{Trili}	90.1 ($\pm$0.08)	95.7 ($\pm$0.09)	**99.3** ($\pm$0.04)

TABLE V: Retrieval-based dialogue: Experimental results on the Ubuntu Corpus V1 corpus. The values associated with $\pm$ in brackets are standard deviation. Mean and standard deviation results are averaged over five runs of each model. Four TAN variants of different tensor operation-based Tri-Attention mechanisms: TAdd - T-Additive, TDP - T-Dot-Product, TSDP - T-Scaled-Dot-Product, and Trili - Trilinear.

- *$BiMPM_{char}$* and *$BiMPM_{word}$*: They employ bilateral multi-perspective matching to determine the semantic consistency between sentences; BiLSTM learns sentence representation, matches two sentences from multi-perspectives, aggregates the matching results, and makes prediction by a dense layer [65].
- *MSEM*: A connected graph describes the relations between sentences and realizes a neural architecture for multi-task learning including both sentence matching and classification [66].
- *GMN*: A neural graph matching network is fed with all possible segmentation paths to form word lattice graphs and learns graph-based representations of sentences [42].
- *COIN*: A cross-attention mechanism combines contextual information aligning sequences, with aligned representations interpolated by a gate fusion layer [13].
- *3DSSM*: A 3D CNN captures temporal and multi-granular features of sentences and generates matching representation [21].

Lastly, we compare TAN to several pretrained neural language networks:

- *BERT-Chinese*: A Chinese BERT model [50].
- *BERT-wwm*: A Chinese BERT with an entire word masking mechanism during pretraining [67].
- *BERT-wwm-ext*: A variant of BERT-wwm with more

training data and steps [67].

- *ERNIE*: It learns language representation enhanced by knowledge masking strategies and entity- and phrase-level masking [68].
- *K-BERT*: A model enhances BERT with HowNet by introducing soft position and visible matrix during fine-tuning and inference phases [69].
- *GMN-BERT*: A neural graph matching network with multi-granular input information [42].

2) TAN Settings: The TAN with Tri-Attention mechanism is customized for sentence semantic matching. Its architecture is composed of four core modules as shown in Fig. 3. Compared to the TAN variant for retrieval-based dialogue, the filtering module is not necessary. This is because the filtering module trims too long sequences, while the sequence length in this task is acceptable. Since the input of sentence semantic matching consists of only two sentences, the contextual feature representation is learned by feeding connected results of question and answer into $BERT_n$ [42].

3) Performance Evaluation: The results of TAN for sentence semantic matching compared with all baseline models are shown in Table VI. We obtain the following observations.

First, TAN significantly outperforms all non-attention classic methods, similar to that for retrieval-based dialogue. The classic networks are unable to effectively capture latent semantic information and complex interaction patterns. In contrast, our Tri-Attention-based TAN combines the advantages of pretrained language model with contextual attention.

Second, TAN also beats advanced networks with interaction or Bi-Attention mechanism. Similar to retrieval-based dialogue, TAN gains advantages from the prior knowledge learned by the pretrained model on large-scale corpus and contextual attention captured by Tri-Attention.

Further, TAN performs better than pretrained BERT variants. The BERT variants including BERT-wwm, BERT-wwm-ext and ERNIE were trained on large data with more subtle training techniques. In contrast, TAN outperforms these baselines without the involvement of large data or specific training techniques. This means that TAN not only outperforms these powerful BERT variants in terms of accuracy and F_1-score but also involves less training costs. Additionally, both K-BERT and GMN-BERT are the latest BERT variants for sentence semantic matching. Their performance is still inferior to the best results of TAN_{TAdd} and TAN_{TSDP}.

Lastly, TAN_{TAdd}, TAN_{TDP}, and TAN_{TSDP} outperform all baselines in terms of accuracy, and TAN_{TAdd} and TAN_{TSDP} achieve better performance than baselines in terms of F_1-score. These results verify the effectiveness of our model. TAN builds on BERT by replacing its self-attention mechanism with Tri-Attention mechanism, whose four similarity implementations make 1-2% improvement in terms of accuracy compared to the original BERT. This indicates the Tri-Attention mechanism plays a core role in making TAN better than BERT, i.e., beating the standard Bi-Attention mechanism.

E. Task 3: Machine Reading Comprehension

The last NLP task for evaluating TAN with Tri-Attention against the baselines is machine reading comprehension. We

Methods	Accuracy	F_1-score
WMD_{char}	70.6	73.4
WMD_{word}	60.0	70.8
CNN_{char}	71.8	75.2
CNN_{word}	72.8	75.7
$BiLSTM_{char}$	73.5	77.5
$BiLSTM_{word}$	76.1	78.9
$BiMPM_{char}$	83.4	85.0
$BiMPM_{word}$	83.3	84.9
MSEM	85.7	-
GMN	84.6	86.0
COIN	85.6	86.5
3DSSM	85.7	86.4
BERT	85.73	86.86
BERT-wwm	86.80	87.78
BERT-wwm-ext	86.68	87.71
ERNIE	87.04	**88.06**
K-BERT	87.10	-
GMN-BERT	**87.30**	88.0
TAN_{TAdd}	**87.49** (±0.47)	**87.95** (±0.27)
TAN_{TDP}	87.25 (±0.11)	87.83 (±0.06)
TAN_{TSDP}	87.23 (±0.58)	87.80 (±0.29)
TAN_{Trili}	86.72 (±0.05)	87.38 (±0.02)

TABLE VI: Sentence semantic matching: Experimental results on the LCQMC data. Mean and standard deviation results are averaged over five runs of each model.

evaluate this in terms of three sets of baselines: single models, ensemble models, and pretrained lanaguage neural networks.

1) Baseline Methods: First, we compare TAN with single model-based methods:

- *Stanford AR*: The standard Bi-Attention mechanism obtains the question-related passage representation, which is then coupled with a candidate option to obtain the relevance score [45].
- *GA Reader*: A gated attention mechanism captures the passage information related to a problem, which is coupled with candidate options [45].
- *ElimiNet*: An elimination gate discards irrelevant options, refines passage representations, and selects the most suitable option by a selection module [70].
- *HAF*: A hierarchical attention mechanism captures the interactions between passages, questions and candidate options [71].
- *MUSIC*: It dynamically applies different matching strategies to different questions and applies a multi-hop reasoning method to reach the right answer [72].
- *Hier-Co-Matching*: A co-matching strategy matches a passage to a question and a passage to a candidate answer, and then leverages a hierarchical LSTM to encode co-matching states for the relevance representation [73].

Second, we compare TAN with ensemble models:

- *GA Reader (6-ensemble)*: An ensemble method based on multi-hop gated attention mechanism between passages and questions [45].
- *ElimiNet (6-ensemble)*: An ensemble method with a multi-hop mechanism eliminates candidate answers [70].
- *GA + ElimiNet (12-ensemble)*: It integrates GA Reader (6-ensemble) and ElimiNet (6-ensemble) [74].

- *Dynamic Fusion Network (9-ensemble)*: An ensemble model based on multi-strategy inference for a comprehension architecture [72].
- *CSA Model + ELMo (9-ensemble)*: It integrates spatial convolution attention mechanism with pretrained language model ELMo [74].

Lastly, we compare TAN with pretrained language models:

- $BERT_{base}$: It is the most commonly used pretrained language model structure [50].
- $BERT_{base} + DCMN$: A reading comprehension model based on $BERT_{base}$ and a dual co-matching network mechanism [6].

2) TAN Settings: We here customize TAN with Tri-Attention for machine reading comprehension. As shown in Fig. 3, its model architecture is similar to that of retrieval-based dialogue. However, this task involves three input sequences. Similar to sentence semantic matching, contextual feature representation is obtained by feeding the concatenation of question, answer, and passage representations into $BERT_n$.

3) Performance Evaluation: The results of TAN with Tri-Attention for machine reading comprehension in comparison with all baselines are reported in Table VII. Again, TAN with Tri-Attention outperforms most baselines, verifying their effectiveness.

First, TAN significantly outperforms all single-model-based methods. This is probably because these single models have a single structure which cannot model the semantic information within a text and the semantic relations between texts. TAN embedded with a pretrained language model and contextual attention shows their empower.

Second, our approach is clearly superior to all ensemble ones. The first four ensemble models: GA Reader (6-ensemble), ElimiNet (6-ensemble), GA + ElimiNet (12-ensemble), and Dynamic Fusion Network (9-ensemble), do not apply pre-training, which may be the main reason for their low performance. The last ensemble model CSA Model + ELMo (9-ensemble) utilizes the pretrained language model ELMo [75] to enhance its performance. Although this model has been greatly improved compared with the previous four models, its performance is still much lower than TAN. This may be due to two reasons: the advantage of pretrained BERT in our model and the contextual enhancement by Tri-Attention.

Lastly, similar to the above conclusions in other tasks, TAN with pretrained BERT beats the original BERT with self-attention by explicitly capturing contextual interactions with queries and keys. Specifically, the TAN with additive Tri-Attention similarity slightly outperform $BERT_{base}$+DCMN, which is specific for machine reading comprehension and relies on strategies and skills used by humans to complete reading comprehension tasks. In contrast, as a general attention mechanism, TAN and Tri-Attention are applicable to machine reading comprehension and other NLP tasks.

F. Bi-Attention vs Tri-Attention Comparison

Here, we further evaluate the effectiveness of Tri-Attention. Two sets of comparisons are undertaken. The first compares Bi-Attention with Tri-Attention to show the effectiveness

Methods	Accuracy
Stanford AR	43.3
GA Reader	44.1
ElimiNet	44.7
HAF	47.2
MUSIC	47.4
Hier-Co-Matching	50.4
GA Reader (6-ensemble)	45.9
ElimiNet (6-ensemble)	46.5
GA + ElimiNet (12-ensemble)	47.2
Dynamic Fusion Network (9-ensemble)	51.2
CSA Model + ELMo (9-ensemble)	55.0
$BERT_{base}$	**65.0**
$BERT_{base}$+DCMN	**67.0**
TAN_{TAdd}	**67.5** (± 0.04)
TAN_{TDP}	66.9 (± 0.13)
TAN_{TSDP}	66.7 (± 0.07)
TAN_{Trili}	66.1 (± 0.14)

TABLE VII: Machine reading comprehension: Experimental results on RACE. Means and standard deviations are averaged over five runs of each model.

of Tri-Attention. The second compares Tri-Attention with a commonly used contextual attention enhancement method in the literature to show the need for capturing explicit query-key-context interactions. As the RACE data costs too much time on running a full set of experiments, we here report the results on LCQMC and the Ubuntu Corpus V1 corpus.

1) Effectiveness: Bi-Attention vs Tri-Attention Mechanisms: We evaluate the different effect of the Bi-Attention without contextual information versus the Tri-Attention with context in the same network structure. The experimental results are shown in Table VIII. Bi-Attention mechanism does not involve contextual information in measuring the interactions. Its other parts are the same as Tri-Attention. We report the results of these two attention mechanisms under four relevance scoring operations: TAdd, TDP, TSDP, and Trili, respectively.

Overall, the results show that the Tri-Attention variants consistently outperform the counterparts without context. This corroborates the contribution of Tri-Attention with context, even though the performance varies over different relevance scoring functions.

Specifically, on LCQMC, the difference between Bi-Attention and Tri-Attention mechanisms shows the greatest under trilinear operation. Tri-Attention gains extra 1.03% accuracy and 0.76% F_1-score over Bi-Attention, respectively. For the Ubuntu Corpus V1 corpus, the greatest difference is associated with SDP, where Tri-Attention makes 0.8%, 0.2%, and 0.1% improvement in terms of evaluation metrics $R_{10}@1$, $R_{10}@2$, and $R_{10}@5$, respectively.

2) Necessity: Contextual Query-Key Interactions vs Query-Key-Context Interactions: As discussed in Section II, some existing methods also involve contextual information, typically by simple addition or concatenation to target representations. Here, we evaluate the difference of this way from our approaches in Tri-Attention which involves context equivalently to other query and key entities in the attention learning.

We generate a Bi-Attention variant to realize a contextual attention enhancement method commonly used in the existing

Methods	LCQMC		Ubuntu Corpus V1 Corpus		
	Accuracy	F_1-score	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
Tri-Attention$_{TAdd}$	**87.49** ($\pm$0.47)	**87.95** ($\pm$0.27)	**90.5** ($\pm$0.3)	**95.8** ($\pm$0.06)	**99.2** ($\pm$0.05)
Bi-Attention$_{Add}$	86.77 ($\pm$0.28)	87.37 ($\pm$0.16)	89.9 ($\pm$0.4)	95.5 ($\pm$0.2)	**99.2** ($\pm$0.09)
Tri-Attention$_{TDP}$	**87.25** ($\pm$0.11)	**87.83** ($\pm$0.06)	**90.3** ($\pm$0.1)	**95.9** ($\pm$0.3)	**99.3** ($\pm$0.06)
Bi-Attention$_{DP}$	86.52 ($\pm$0.09)	87.20 ($\pm$0.06)	89.8 ($\pm$0.2)	95.5 ($\pm$0.1)	99.2 ($\pm$0.04)
Tri-Attention$_{TSDP}$	**87.23** ($\pm$0.58)	**87.80** ($\pm$0.29)	**90.6** ($\pm$0.3)	**95.7** ($\pm$0.05)	**99.2** ($\pm$0.04)
Bi-Attention$_{SDP}$	86.61 ($\pm$0.09)	87.29 ($\pm$0.04)	89.8 ($\pm$0.3)	95.5 ($\pm$0.2)	99.1 ($\pm$0.05)
Tri-Attention$_{Trili}$	**86.72** ($\pm$0.05)	**87.38** ($\pm$0.02)	**90.1** ($\pm$0.08)	**95.7** ($\pm$0.09)	**99.3** ($\pm$0.04)
Bi-Attention$_{Bili}$	85.84 ($\pm$0.11)	86.72 ($\pm$0.06)	89.7 ($\pm$0.09)	95.5 ($\pm$0.1)	99.2 ($\pm$0.03)

TABLE VIII: Results of effectiveness of Bi-Attention vs Tri-Attention mechanisms on LCQMC and Ubuntu Corpus V1 Corpus. Means and standard deviations are averaged over five runs.

Methods	LCQMC		Ubuntu Corpus V1 Corpus		
	Accuracy	F_1-score	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
Tri-Attention$_{TAdd}$	**87.49** ($\pm$0.47)	**87.95** ($\pm$0.27)	**90.5** ($\pm$0.3)	**95.8** ($\pm$0.06)	**99.2** ($\pm$0.05)
C-BiAttention$_{Add}$	86.57 ($\pm$0.15)	87.23 ($\pm$0.07)	90.1 ($\pm$0.4)	95.7 ($\pm$0.3)	**99.2** ($\pm$0.06)
Tri-Attention$_{TDP}$	**87.25** ($\pm$0.11)	**87.83** ($\pm$0.06)	**90.3** ($\pm$0.1)	**95.9** ($\pm$0.3)	**99.3** ($\pm$0.06)
C-BiAttention$_{DP}$	85.94 ($\pm$0.21)	86.81 ($\pm$0.12)	90.1 ($\pm$0.3)	95.6 ($\pm$0.2)	99.2 ($\pm$0.06)
Tri-Attention$_{TSDP}$	**87.23** ($\pm$0.58)	**87.80** ($\pm$0.29)	**90.6** ($\pm$0.3)	**95.7** ($\pm$0.05)	99.2 ($\pm$0.04)
C-BiAttention$_{SDP}$	86.64 ($\pm$0.11)	87.32 ($\pm$0.06)	90.2 ($\pm$0.1)	95.7 ($\pm$0.2)	**99.3** ($\pm$0.04)
Tri-Attention$_{Trili}$	**86.72** ($\pm$0.05)	**87.38** ($\pm$0.02)	90.1 ($\pm$0.08)	95.7 ($\pm$0.09)	**99.3** ($\pm$0.04)
C-BiAttention$_{Bili}$	85.94 ($\pm$0.10)	86.82 ($\pm$0.06)	**90.2** ($\pm$0.4)	95.7 ($\pm$0.2)	99.2 ($\pm$0.05)

TABLE IX: Results of contextual query-key interactions vs query-key-context interactions on LCQMC and Ubuntu Corpus V1 Corpus. Means and standard deviations are averaged over five runs.

work. The contextual information is added (with addition) to each sequence separately to obtain a contextual sequence representation. Then, the context-enhanced sequence representations interact using the standard query-key Bi-Attention mechanism, whose other network settings are consistent with Tri-Attention. This forms the commonly used contextual attention in the literature, abbreviated *C-BiAttention*.

The experimental results are shown in Table IX. It shows that the performance of C-BiAttention consistently underperforms than Tri-Attention with all four relevance calculators. C-BiAttention with the dot-product-based relevance calculator achieves the lowest performance. On LCQMC, C-BiAttention reduces 1.52% accuracy and 1.17% F_1-score over Tri-Attention, respectively. Similarly, on the Ubuntu Corpus V1 corpus, the highest performance degradation is with the Additive-based approach. In comparison with Tri-Attention, C-BiAttention decreases 0.4% $R_{10}@1$, 0.1% $R_{10}@2$, respectively. This experiment verifies that the explicit query-key-context interactions in Tri-Attention outperforms the simple contextual attention by adding or concatenating contextual information to underlying sequences. It also confirms the necessity of learning query-key-context interactions in Tri-Attention.

G. Hyperparameter Analysis: Tri-Attention Layers

TAN in Fig. 3 is a stackable structure, where the number of Tri-Attention layers can be dynamically adjusted according to learning tasks. In addition, In addition, different relevance score calculation methods in Eqs. (10)-(13) also affect the number of Tri-Attention layers for a learning task. We thus test the effect of the number of Tri-Attention layers.

Fig. 4 - 6 show the experimental results of retrieval-based dialogue, sentence semantic matching, and machine reading comprehension tasks on Ubuntu Corpus V1, LCQMC, and RACE, respectively. The best performance of Tri-Attention corresponds to different relevance score calculation methods for different tasks. This indicates that, when applying Tri-Attention to a special task, all four relevance calculation methods should be tried before achieving the best result. In addition, the number of Tri-Attention layers depends on the relevance calculation methods and learning tasks. Thus, when applied to a specific task, the best number of Tri-Attention layers needs to be tuned for each relevance calculation method.

H. TAN Case Study

Lastly, we illustrate the prediction results of TAN with Tri-Attention on the LCQMC data for sentence semantic matching. We only illustrate the additive-based relevance calculation for Tri-Attention as the previous experiments show its better stability. We reapply the three variants used in Section VI-F: Tri-Attention$_{Add}$, Bi-Attention$_{Add}$ and C-BiAttention$_{Add}$, respectively. To verify the advantages of Tri-Attention, the prediction results of all three models were statistically analyzed.

Specifically, we conduct a statistical analysis when only Tri-Attention makes correct prediction while the other two models predict incorrectly. In the test set of LCQMC, there are 151 pieces of data conforming to the above situation, among which 49 pieces are positive and 102 pieces are negative. This suggests that Tri-Attention is better at identifying negative data. The instance in Table X illustrates the result.

For both S_1 and S_2, the gold label is 0, which means that their meanings are different. Our Tri-attention can make a
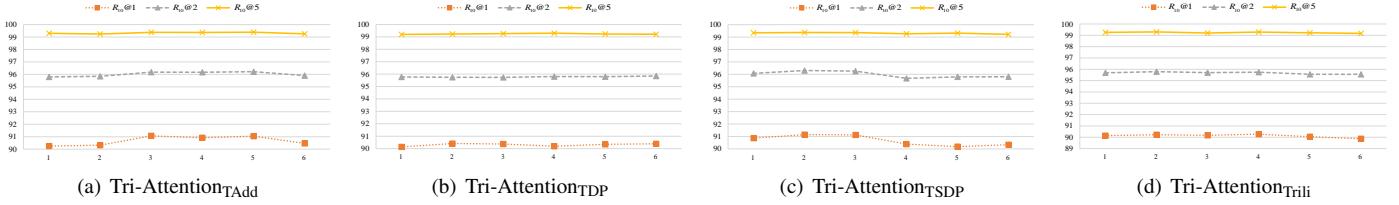
Fig. 4: Performance comparison on the Ubuntu Corpus V1 corpus with different number of Tri-Attention layers.
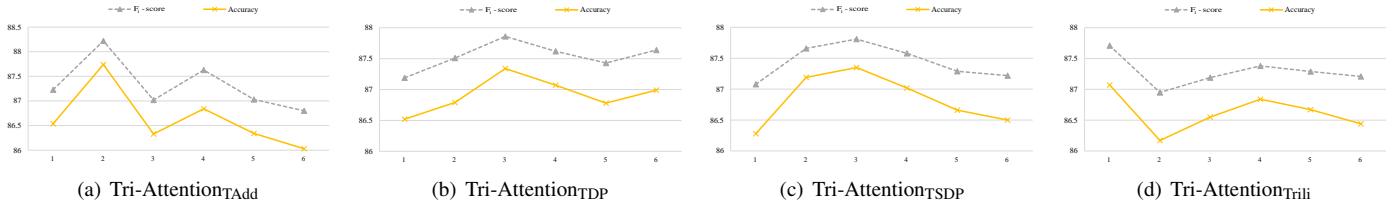
(a) Tri-Attention$_{TAdd}$　(b) Tri-Attention$_{TDP}$　(c) Tri-Attention$_{TSDP}$　(d) Tri-Attention$_{Trili}$

Fig. 5: Performance comparison on LCQMC with different number of Tri-Attention layers.

(a) Tri-Attention$_{TAdd}$　(b) Tri-Attention$_{TDP}$　(c) Tri-Attention$_{TSDP}$　(d) Tri-Attention$_{Trili}$

Fig. 6: Performance comparison on RACE with different number of Tri-Attention layers.

(a) Tri-Attention$_{TAdd}$　(b) Tri-Attention$_{TDP}$　(c) Tri-Attention$_{TSDP}$　(d) Tri-Attention$_{Trili}$

	Data	Label
S_1	哪些浏览器可以看电影 (En: Which browsers can play movies)	0
S_2	什么浏览器可以下电影 (En: Which browsers can download movies)	

TABLE X:　Case study examples from LCQMC.

correct judgement, while the others cannot. Since there are many overlap words in S_1 and S_2, if a model judges the relations between these two sentences only on the basis of word-level relevance without the contextual information, it is easy to be misled and draw a wrong conclusion. This probably explains why the Bi-Attention$_{Add}$ method fails to correctly predict the relation between S_1 and S_2. As for C-BiAttention$_{Add}$, although it also involves context, its contextual information does not participate in the interaction with queries and keys, thus making no direct impact on the relevance score between sentences. While Tri-Attention directly involves context in the interactions between sentences, contributing to better prediction.

VII. CONCLUSIONS

Contextual information has shown essential in many learning tasks. The great success of attention mechanisms does not necessarily involves contextual information. In neural NLP, increasing work on contextual attention learning typically concatenates contextual features into underlying targets such as sequences, then the standard query-key-based Bi-Attention mechanisms calculate relevance scores on the tokenized contextual sequence representations. In this paper, a novel query-key-context-interactive Tri-Attention mechanism explicitly captures the interactions between query, key and context. We derive four query-key-context relevance calculation methods for Tri-Attention using tensor algebraic techniques. Intensive experiments on different NLP tasks show that Tri-Attention-based networks can serve as a general attention framework, which outperforms most state-of-the-art non-attention, standard Bi-Attention, contextual Bi-Attention, and pretrained language models with attention. Our future work includes evaluating Tri-Attention in other NLP tasks and Tri-Attention without pretrained BERT, and exploring more effective tensor algebraic implementations for for interactions with $n > 3$ factors.

REFERENCES

[1] K. Oberauer, "Working memory and attention - A conceptual analysis and review," *Journal of Cognition*, vol. 2, no. 1, p. 36, 2019.

[2] D. Hu, "An introductory survey on attention mechanisms in NLP problems," in *Proceedings of SAI Intelligent Systems Conference*, 2019, pp. 432–448.

[3] A. Raganato, Y. Scherrer, and J. Tiedemann, "Fixed encoder self-attention patterns in transformer-based machine translation," in *EMNLP*, 2020, pp. 556–568.

[4] B. Lyu, L. Chen, S. Zhu, and K. Yu, "LET: Linguistic knowledge enhanced graph transformer for Chinese short text matching," in *AAAI*, 2021, pp. 13 498–13 506.

[5] Z. Ye, Y. Qin, and W. Xu, "Financial risk prediction with multi-round Q&A attention network," in *IJCAI*, 2020, pp. 4576–4582.

[6] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, and X. Zhou, "DCMN+: Dual co-matching network for multi-choice reading comprehension," in *AAAI*, 2020, pp. 9563–9570.

[7] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, "Decoupled attention network for text recognition," in *AAAI*, 2020, pp. 12 216–12 224.

[8] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "Abcdm: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, no. 115, pp. 279–294, 2021.

[9] L. Ding, L. Wang, and D. Tao, "Self-attention with cross-lingual position representation," in *ACL*, 2020, pp. 1679–1685.

[10] C. Gilbert, M. Ito, M. Kapadia, and G. Westheimer, "Interactions between attention, context and learning in primary visual cortex," *Vision Research*, vol. 40, no. 10, pp. 1217–1226, 2000.

[11] P. Knoeferle, "How context influences language processing and comprehension," *Research Outreach*, 2020.

[12] R. Willems and M. Peelen, "How context changes the neural basis of perception and language," *iScience*, vol. 24, no. 5, p. 102392, 2021.

[13] Z. Hu, Z. Fu, Y. Yin, and G. de Melo, "Context-aware interaction network for question matching," in *EMNLP*, 2021, pp. 3846–3853.

[14] S. Storks, Q. Gao, and J. Y. Chai, "Recent advances in natural language inference: A survey of benchmarks, resources, and approaches," *arXiv preprint arXiv:1904.01172*, 2019.

[15] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, "Simple and effective text matching with richer alignment features," in *ACL*, 2019, pp. 4699–4709.

[16] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *NIPS*, 2014, pp. 2204–2212.

[17] J. Zeng, S. Wu, Y. Yin, Y. Jiang, and M. Li, "Recurrent attention for neural machine translation," in *EMNLP*, 2021, pp. 3216–3225.

[18] T. Zhang, H. Huang, C. Feng, and L. Cao, "Enlivening redundant heads in multi-head self-attention for machine translation," in *EMNLP*, 2021, pp. 3238–3248.

[19] T. Zhang, H. Huang, L. Cao, and C. Feng, "Self-supervised bilingual syntactic alignment for neural machine translation," in *AAAI*, 2021, pp. 14 454–14 462.

[20] Y. Guan, Z. Li, Z. Lin, Y. Zhu, J. Leng, and M. Guo, "Block-Skim: Efficient question answering for Transformer," in *AAAI*, 2022, pp. 10 710–10 719.

[21] W. Lu, R. Yu, S. Wang, C. Wang, P. Jian, and H. Huang, "Sentence semantic matching based on 3D CNN for human-robot language interaction," *ACM Transactions on Internet Technology*, vol. 21, no. 4, pp. 98:1–98:24, 2021.

[22] G. Zhang, W. Lu, X. Peng, S. Wang, B. Kan, and R. Yu, "Word sense disambiguation with knowledge-enhanced and local self-attention-based extractive sense comprehension," in *COLING*, 2022.

[23] A. Rashed, S. Elsayed, and L. Schmidt-Thieme, "Context and attribute-aware sequential recommendation via cross-attention," in *RecSys*, 2022, pp. 71–80.

[24] Y. Zheng and G. Florez Arias, "A family of neural contextual matrix factorization models for context-aware recommendations," in *UMAP*, 2022, pp. 1–6.

[25] Q. Zhang, L. Cao, C. Shi, and Z. Niu, "Neural time-aware sequential recommendation by jointly modeling preference dynamics and explicit feature couplings," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[26] L. Cao, "Beyond IID: Non-IID thinking, informatics, and learning," *IEEE Intelligent Systems*, vol. 37, no. 4, pp. 5–17, 2022.

[27] L. Cao and C. Zhu, "Personalized next-best action recommendation with multi-party interaction learning for automated decision-making," *Plos one*, vol. 17, no. 1, p. e0263010, 2022.

[28] C. Zhu, L. Cao, and J. Yin, "Unsupervised heterogeneous coupling learning for categorical representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 533–549, 2020.

[29] B. Yang, J. Li, D. F. Wong, L. S. Chao, X. Wang, and Z. Tu, "Context-aware self-attention networks," in *AAAI*, 2019, pp. 387–394.

[30] L. Ding, L. Wang, D. Wu, D. Tao, and Z. Tu, "Context-aware cross-attention for non-autoregressive translation," in *COLING*, 2020, pp. 4396–4402.

[31] Y. Zeng, Z. Lin, H. Lu, and V. M. Patel, "CR-Fill: Generative image inpainting with auxiliary contextual reconstruction," in *ICCV*, 2021, pp. 14 164–14 173.

[32] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, and H. Yu, "Deep learning for image inpainting: A survey," *Pattern Recognition*, p. 109046, 2022.

[33] P. Song, D. Guo, J. Cheng, and M. Wang, "Contextual attention network for emotional video captioning," *IEEE Transactions on Multimedia*, 2022.

[34] Y. Pang, L. Wu, Q. Shen, Y. Zhang, Z. Wei, F. Xu, E. Chang, B. Long, and J. Pei, "Heterogeneous global graph neural networks for personalized session-based recommendation," in *WSDM*, 2022, pp. 775–783.

[35] A. Hallak, D. D. Castro, and S. Mannor, "Contextual Markov decision processes," *ArXiv Preprint*, p. arXiv:1502.02259, 2015.

[36] C. Benjamins, T. Eimer, F. Schubert, A. Mohan, A. Biedenkapp, B. Rosenhahn, F. Hutter, and M. Lindauer, "Contextualize me - The case for context in reinforcement learning," *ArXiv Preprint*, p. arXiv:2202.04500, 2022.

[37] J. Liu, M. Gong, Z. Tang, A. Qin, H. Li, and F. Jiang, "Deep image inpainting with enhanced normalization and contextual attention," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[38] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015, pp. 1–15.

[40] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015, pp. 1412–1421.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[42] L. Chen, Y. Zhao, B. Lyu, L. Jin, Z. Chen, S. Zhu, and K. Yu, "Neural graph matching networks for Chinese short text matching," in *ACL*, 2020, pp. 6152–6158.

[43] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," in *SIGDIAL*, 2015, pp. 285–294.

[44] X. Liu, Q. Chen, C. Deng, H. Zeng, J. Chen, D. Li, and B. Tang, "LCQMC: A large-scale Chinese question matching corpus," in *COLING*, 2018, pp. 1952–1962.

[45] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. H. Hovy, "RACE: Large-scale reading comprehension dataset from examinations," in *EMNLP*, 2017, pp. 785–794.

[46] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *ACL*, 2019, pp. 3651–3657.

[47] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using Siamese bert-networks," in *EMNLP*, 2019, pp. 3980–3990.

[48] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[49] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019, pp. 1–18.

[50] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.

[51] R. Kadlec, M. Schmid, and J. Kleindienst, "Improved deep learning baselines for Ubuntu corpus dialogs," *CoRR*, vol. abs/1510.03753, 2015.

[52] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, "Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots," in *ACL*, 2017, pp. 496–505.

[53] Z. Zhang, J. Li, P. Zhu, H. Zhao, and G. Liu, "Modeling multi-turn conversation with deep utterance aggregation," in *COLING*, 2018, pp. 3740–3752.

[54] X. Zhou, L. Li, D. Dong, Y. Liu, Y. Chen, W. X. Zhao, D. Yu, and H. Wu, "Multi-turn response selection for chatbots with deep attention matching network," in *ACL*, 2018, pp. 1118–1127.

[55] C. Tao, W. Wu, C. Xu, W. Hu, D. Zhao, and R. Yan, "One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues," in *ACL*, 2019, pp. 1–11.

[56] Q. Chen and W. Wang, "Sequential attention-based network for noetic end-to-end response selection," *CoRR*, vol. abs/1901.02609, 2019.

[57] C. Yuan, W. Zhou, M. Li, S. Lv, F. Zhu, J. Han, and S. Hu, "Multi-hop selector network for multi-turn response selection in retrieval-based chatbots," in *EMNLP*, 2019, pp. 111–120.

[58] J. Lu, X. Ren, Y. Ren, A. Liu, and Z. Xu, "Improving contextual language models for response retrieval in multi-turn conversation," in *SIGIR*, 2020, pp. 1805–1808.

[59] T. Whang, D. Lee, C. Lee, K. Yang, D. Oh, and H. Lim, "An effective domain adaptive post-training method for BERT in response selection," in *INTERSPEECH*, 2020, pp. 1585–1589.

[60] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *ICML*, 2019, pp. 2790–2799.

[61] J. Gu, T. Li, Q. Liu, Z. Ling, Z. Su, S. Wei, and X. Zhu, "Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots," in *CIKM*, 2020, pp. 2041–2044.

[62] T. Whang, D. Lee, D. Oh, C. Lee, K. Han, D. Lee, and S. Lee, "Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection," in *AAAI*, 2021, pp. 14 041–14 049.

[63] R. Xu, C. Tao, D. Jiang, X. Zhao, D. Zhao, and R. Yan, "Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues," in *AAAI*, 2021, pp. 14 158–14 166.

[64] Y. Li, C. Xu, H. Hu, L. Sha, Y. Zhang, and D. Jiang, "Small changes make big differences: Improving multi-turn response selection in dialogue systems via fine-grained contrastive learning," *CoRR*, 2021.

[65] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *IJCAI*, 2017, pp. 4144–4150.

[66] Q. Huang, J. Bu, W. Xie, S. Yang, W. Wu, and L. Liu, "Multi-task sentence encoding model for semantic retrieval in question answering systems," in *IJCNN*, 2019, pp. 1–8.

[67] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese BERT," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3504–3514, 2021.

[68] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "ERNIE: Enhanced representation through knowledge integration," *arXiv preprint arXiv:1904.09223*, 2019.

[69] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: Enabling language representation with knowledge graph," in *AAAI*, 2020, pp. 2901–2908.

[70] S. Parikh, A. Sai, P. Nema, and M. M. Khapra, "ElimiNet: A model for eliminating options for reading comprehension with multiple choice questions," in *IJCAI*, 2018, pp. 4272–4278.

[71] H. Zhu, F. Wei, B. Qin, and T. Liu, "Hierarchical attention flow for multiple-choice reading comprehension," in *AAAI*, 2018, pp. 6077–6085.

[72] Y. Xu, J. Liu, J. Gao, Y. Shen, and X. Liu, "Towards human-level machine reading comprehension: Reasoning and inference with multiple strategies," *CoRR*, 2017.

[73] S. Wang, M. Yu, J. Jiang, and S. Chang, "A co-matching model for multi-choice reading comprehension," in *ACL*, 2018, pp. 746–751.

[74] Z. Chen, Y. Cui, W. Ma, S. Wang, and G. Hu, "Convolutional spatial attention model for reading comprehension with multiple-choice questions," in *AAAI*, 2019, pp. 6276–6283.

[75] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018, pp. 2227–2237.

Wenpeng Lu received the Ph.D. degree in computer applications from the Beijing Institute of Technology, Beijing, China. He is a professor with the Department of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include natural language processing, machine learning, and their enterprise applications. He is a member of IEEE, CCF and CIPS. He may be contacted at wenpeng.lu@qlu.edu.cu.

Longbing Cao received a Ph.D. degree in pattern recognition and intelligent systems and another Ph.D. in computing sciences. He is a professor and an ARC Future Fellow (level 3) at the University of Technology Sydney. His research interests include AI, data science, machine learning, behavior informatics, and enterprise innovation. He is the EICs of IEEE Intelligent Systems and Springer's JDSA. He may be contacted at longbing.cao@uts.edu.au.

Rui Yu received the master's degree in computer application technology from the Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. He is pursuing the Ph.D. degree at Huazhong University of Science and Technology, Wuhan, China. His research interests include text semantic matching and question answering system. He may be contacted at rui.yu1996@foxmail.com.

Yifeng Li received the Ph.D. in Computer Science from the University of Windsor, Canada. He is an assistant professor and Canada Research Chair (Tier 2) in Machine Learning for Biomedical Data Science at the Department of Computer Science, Department of Biological Sciences, and Centre for Biotechnology, Brock University. His research interests include neural networks, machine learning, data science, optimization, bioinformatics, and chemoinformatics. He may be contacted at yli2@brocku.ca.