Complex & Intelligent Systems (2023) 9:6191–6209
https://doi.org/10.1007/s40747-023-01067-7

ORIGINAL ARTICLE

# Enhancing context representations with part-of-speech information and neighboring signals for question classification

Peizhu Gong[1] · Jin Liu[1] · Yurong Xie[1] · Minjie Liu[1] · Xiliang Zhang[1]

Received: 6 July 2022 / Accepted: 1 April 2023 / Published online: 26 April 2023


## Abstract

Question classification is an essential task in question answering (QA) systems. An effective and efficient question classification model can not only restrict the search space for answers, but also guide the QA system in selecting the optimal knowledge base and search strategy. In recent years, self-attention mechanism has been widely used in question classification for its strength of capturing global dependencies. However, it models all signals with weighted averaging, which is prone to overlooking the relation of neighboring signals. Furthermore, recent research has revealed that part-of-speech (POS) information can be used to determine and reinforce the semantics in sentence representation. In this paper, we propose a POS-aware adjacent relation attention network (POS-ARAN) for question classification, which enhance context representations with POS information and neighboring signals. To consider the local context, we propose an adjacent relation attention mechanism, which incorporates a Gaussian bias via a dynamic window to revise the vanilla self-attention mechanism. Thus, it can capture both the long-term dependency and local representation of semantic relations among words in different sentences. In addition, a POS-aware embedding layer is proposed, which helps to locate the appropriate headwords by syntactic information. Extensive experiments are conducted on Experimental Data for Question Classification (EDQC) dataset and Yahoo! Answers Comprehensive Questions and Answers 1.0, the results demonstrate that our model significantly outperforms the existing methods, achieving 95.59% in coarse-grained level accuracy and 92.91% in fine-grained level accuracy, respectively.

**Keywords** Question classification · Self-attention mechanism · Adjacent relation · Part-of-speech · Context representation

## Introduction

Question classification is an essential task in the natural language understanding module of question answering (QA), which aims to classify questions into certain pre-defined intent categories. Previous studies [1–3] indicate that an efficient question classification module contributes to restricting the search space for finding answers, thus reducing search costs. A typical example is "What's the capital of China?". The real intent of this question is "locations". Therefore, the candidate of answers is the "locations" pronoun related to "China" and "capital", which is a much smaller space than the entire search. Moreover, an efficient question classification module also can guide the QA system to select the optimum knowledge base and search strategy. For example, "Who is the CEO of Facebook", focusing more on the relation among different entities, is well suited for searching answers in a knowledge graph. While if the question is "Why Facebook will never charge you?", whose intent is "reasons", the best choice is to search for answers in a web knowledge base.

In the early stage, many rule-based methods [4, 5] tried to match questions with hand crafted templates to determine the category of question. However, plenty of rules need to manually pre-defined for different cases, which is time-consuming and labor-intensive. With the rapid development of deep learning and lexical embedding techniques, neural networks have made great breakthroughs in natural lan-

✉ Jin Liu
jinliu@shmtu.edu.cn

Peizhu Gong
gongpeizhu012@163.com

Yurong Xie
xieyurong@shmtu.edu.cn

Minjie Liu
201730310085@stu.shmtu.edu.cn

Xiliang Zhang
azxl1997@163.com

[1] College of Information Engineering, Shanghai Maritime University, Shanghai, China

guage processing, especially convolutional neural networks (CNN) [6, 7] and recurrent neural networks (RNN) [8–10]. RNN-based models view text as a sequence of words, which are intended to capture word dependencies in text. Bai et al. [11] proposed a positional RNN model that considers aspect word position information for text classification. Therasa et al. [12] introduced an adaptive RNN model with feature optimization to learn question representation with faster convergence and avoid the local optima. Both of them have achieved a great improvement in results. CNN-based models are trained to identify textual patterns, such as key phrases and text structure. Soni et al. [13] presented a CNN-based architecture that applies two-dimensional multi-scale convolutional procedures to extract intra- and inter-sentence features from input text data. Tan et al. [14] proposed an adaptive CNN model that adaptively generated convolutional filters to project word embeddings into the same subspace. In addition, hybrid models [15] combining CNNs and RNNs are also proposed. Ma et al. [16] put forward a hierarchical convolutional recurrent neural network for Chinese question classification, which combines TextCNN and Bi-LSTM to learn human-understandable concepts in a hierarchical structure. However, RNNs are susceptible to gradient disappearance or explosion and have high time complexity due to their recursive nature, While CNNs do not consider sequence order and fail to capture long-range dependencies. In the recent years, self-attention mechanism [17, 18] has been widely used in question classification for its strength of capturing global dependencies. Liu et al. [19] proposed a multi-stage attention model based on temporal CNN, capturing contextual-related features at word and concept levels. Zheng et al. [20] constructed a deep learning model combining RNNs and attention mechanism, in which RNNs generated the semantic features, and the obtained features were weighed in accordance with the attention mechanism. However, vanilla self-attention mechanism models all signals with weighted averaging, which is prone to overlooking the relation of neighboring signals [21]. Moreover, recent studies [22, 23] have demonstrated that fully utilizing part-of-speech (POS) information would result in additional semantic improvements in sentence representations.

Accordingly, in this paper, we propose a POS-aware adjacent relation attention network (POS-ARAN) for question classification, which enhance context representations with POS information and neighboring signals. Specifically, we propose an adjacent relation attention mechanism (ARAM), which revise the vanilla self-attention mechanism by incorporating a Gaussian bias via a dynamic non-symmetrical window. In this way, additional contextual information between neighboring words can be captured, while long-term dependencies are not affected. In addition, a POS-aware embedding layer is proposed, which helps to locate the appropriate headwords by syntactic information. Extensive experiments are conducted on Experimental Data for Question Classification (EDQC) dataset and Yahoo! Answers Comprehensive Questions and Answers 1.0, the results demonstrate that our model significantly improves the performance, achieving 95.59% in coarse-grained level accuracy and 92.91% in fine-grained level accuracy, respectively.

Our contributions are summarized as follows:

- We propose an adjacent relation attention mechanism (ARAM). The ARAM revises the original self-attention mechanism by integrating a learnable Gaussian bias within a dynamic window, which enables it to capture additional contextual information between neighboring words.
- We propose a POS-aware embedding layer, which helps to locate the appropriate headwords by syntactic information for textual content understanding.
- We conduct our experiments on the widely used question classification datasets, and the experimental results show that our proposed models can achieve better performance than previous state-of-the-art models.

The remaining of this article is structured as follows: In Sect. "Related work", relevant studies about this task are presented. In Sect. "Method", we formalize the definition of question classification. Moreover, the method and architecture of POS-ARAN are completely described in this section. In Sects. "Experiments" and "Discussion", we present the results of the comparison experiments and ablation experiments to prove that our model is competitive and provide a brief analysis. Finally, in Section. "Conclusion", some conclusions are summarized.

## Related work

Question Answering (QA) is a vital task in natural language processing (NLP) [24, 25]. The goal is to build systems that can automatically answer questions raised by human in a natural language [26]. Question classification is a key sub-task to QA systems, which aims to map the question into a certain category. Various techniques have emerged to solve question classification issues. These techniques can be divided into three groups: rule-based techniques, machine learning techniques, and deep learning techniques.

### Rule-based approaches

Initially, most question classifiers followed rule-based strategies, such as the Webclopedia QA Typology [27], including 276 hand-written rules corresponding to the 180 answer types. Dragomir et al. [28] employ Ripper and a heuristic rule-based algorithm to identify the question type. Kwok et

al. [29] introduced MULDER that can determine the question type just by looking at the questions interrogative pronoun. Silva et al. [30] evaluated a rule-based question classifier, either it directly matched the question classification or it identified the headword of questions and mapped it into the question category by WordNet. Although rule-based methods can achieve good results in specific domains without the need for large amounts of training data, they have been phased out as different expressions can significantly increase the number of rule templates and consume a lot of resources and time.

### Machine-learning-based approaches

With the development of machine learning, approaches applied in question classification have gradually changed away from manual and expert rules. Most of these methods are based on supervised statistical machine learning. Huang et al. [31] used the support vector machine (SVM) with linear kernel to classify questions and achieved the accuracy of 89.2% on Text REtrieval Conference (TREC) dataset. Zhang et al. [32] designed a kind of SVM with a tree-like custom kernel, which enabled their model to achieve the accuracy of 90.0% on TREC dataset. In addition to the SVM model, researchers also employed the maximum entropy model in question classification. Kocik et al. [33] proposed a maximum entropy model on TREC and achieved the accuracy of 89.8%. Le Nguyen et al. [34] put forward a subtree mining algorithm that uses subtrees of parse trees as features and combines a maximum entropy model to classify questions. Furthermore, the Sparse Network (SNoW) is also an alternative, which trains an independent linear function for each class by updating the rules. Li et al. [35] introduced a hierarchical classifier and firstly allocated crude tags to questions. These tags and other features are then entered into the subsequent hierarchy for classification. They attained an accuracy of 89.3% on the TREC dataset.

### Deep-learning-based approaches

Compared with traditional machine learning, one of the improvements of deep learning lies in the word embedding. Traditional word embedding models like the N-GRAM are prone to data smoothing due to sparse data. Furthermore, the word vector obtained from such a model can reflect neither the diversity of words nor the connection between the words. In contrast, distributed representation methods like word2vec utilize low-dimensional, dense vectors to represent the semantic information of words, effectively solving the above problem. Yilmaz et al. [36]used word2vec on different deep learning architectures to study an agglutinative language, achieving better performance than traditional methods on multiple datasets. Although some studies still used n-gram features, they tried to employ CNNs to compensate for the shortcomings. Kim et al. [37] proposed a CNN-based classification model to capture the n-grams features of text. This model is simple and consists of only five layers. Soni et al. [13] presented a CNN-based architecture that applies two-dimensional multi-scale convolutional procedures to extract intra- and inter-sentence features from input text data. Tan et al. [14] proposed an adaptive CNN model that adaptively generated convolutional filters to project word embeddings into the same subspace. RNN is also a widely used approach to capture word dependencies in text. Zhou et al. [38] used 2D convolution and 2D pooling layer to obtain the representation of output from Bi-LSTM. In this way, their model could capture more sequence features and vector features simultaneously. Similarly, Wu et al. [39] introduced two Bi-LSTM to generate hidden state representations of the question and answer text respectively. Cai et al. [40] proposed a classification model based on CNN-LSTM network, which focused on the medical field. The experimental results on Health Care Quality Indicators (HCQI) dataset proved the efficiency of their model. RNN model that considers aspect word position information for text classification. Therasa et al. [12] introduced an adaptive RNN model with feature optimization to learn question representation with faster convergence and avoid the local optima. With the occurrence of Transformer, the self-attention mechanism has become a matter of concern. It can generate the weights of different connections dynamically, enabling it to handle the long-term dependency in sentences. Liang et al. [41] proposed a novel SVA-CNN deep learning architecture, which leveraged a multi-view representation of text to learn high-level features. Simultaneously, spatial attention and view attention mechanisms were proposed to preserve the latent interaction among different-granularity semantic groups. Liu et al. [19] proposed a multi-stage attention model based on temporal CNN, capturing contextual-related features at word and concept levels. Zheng et al. [20] constructed a deep learning model combining RNNs and attention mechanism, in which RNNs generated the semantic features, and the obtained features were weighed in accordance with the attention mechanism.

However, since the conventional self-attention models consider all words in a sequence, so that the relation among neighboring words is weakened in the weighted averaging. As we all know, neighboring words frequently include a wealth of information about the words that are close to them, which is crucial for models to understand the semantics of natural language. To solve this problem, POS-ARAN, which incorporates POS information and enhances the model's comprehension of local context, is suggested in this paper.

## Method

### Problem definition

Question classification can be defined as follows. The initial input is a question $Q$

$$Q = \{q_1, q_2, ..., q_m\}, \tag{1}$$

where $m$ denotes the number of words in the input question. Then, POS-ARAN performs a nonlinear transformation on $Q$ and gains the result $\mathcal{R}$

$$\mathcal{R} = \{r_1, r_2, ...r_n\}, \tag{2}$$

where $n$ denotes the number of intent categories. $\mathcal{R}$ can be regarded as the predicted scores for $n$ intent categories. Finally, POS-ARAN selects the category with the highest score $r_i$ as the outputted intent category

$$r_i = max\{r_1, r_2, ...r_n\}, i \in [1, 2, ..., n]. \tag{3}$$

### Overall architecture

In this section, we introduce the overall architecture of our POS-ARAN model which is shown in Fig. 1. The model first receives an input question and transforms the sequence into a word vector matrix. The POS-aware embedding layer presented in this paper is based on GloVe, which has been pretrained on Wikipedia 2014 and Gigaword 5, containing over 6 billion tokens. By calculating the similarity of the original word to the rest of the words in the sentence, the alignment probabilities as attention values are generated. Among them, the word having the maximum alignment probability is defined as the headword. Then, the word vector matrix is sent to the ARAM module, which is the main module in POS-ARAN. It applies revised self-attention to capture the long-term dependency of the headword. During this process, the adjacent relation of each word is encoded to improve the short-term dependency of the headword. Finally, the encoded matrix is passed to a softmax layer and obtains the final probability of categories.

### Adjacent relation attention mechanism

As shown in Fig. 2, the original distribution of attention treats the same words at various distances almost equally, which is inconsistent with human's cognition of natural language texts that the neighboring words of headword could provide more richer semantic information. Namely, we hope the original word $x_i$ can keep highly relevant to the neighboring words of the headword $x_j$ when $x_i$ is aligned with $x_j$. In this way, the long-term dependency and local information can be kept at the same time.

The conventional self-attention takes all of the words in the sentence into consideration, so that the weighted averaging weakens the short-term dependency between the adjacent words. Hence, we introduce the idea that revising the original attention distribution to take into account the expected local information. In addition, the ideal distribution should only correct the distribution of some necessary positions, which could avoid unnecessary interference on the original distribution.

In this paper, we hypothesize that the semantic contribution to headword from tokens at different distance obeys a normal distribution. The reason to choose normal distribution is that it is hard to statistically measure the semantic importance of a word according to another one. Compared with different decaying mechanisms, e.g., linear decaying according to distance [42], or other distributions such as Zipf Distribution, studies demonstrate that the Gaussian [43] assumption works better. Specifically, we learn a couple of Gaussian biases with a non-symmetrical window instead of the symmetrical window, and add these biases to the attention distribution. As shown in Fig. 2, when self-attention aligns "Where" with "capital", it should not only focus on "capital" but also on the words adjacent to "capital". Figure 2a applies the traditional symmetrical window whose center is "capital". However, not all words in the window are related to the headword. For example, "is" and "the" nearly have nothing to do with "capital" and should not receive much attention. Directly using a symmetrical window may pay attention to some irrelevant words. As shown in Fig. 2b, POS-ARAN applies the non-symmetrical window to correct the distribution. It is obvious that more attention is paid to "of" and "China", which is more reasonable.

Given an input sequence $X$=$\{x_1, x_2, ..., x_n\}$, self-attention will map it to a target sequence $H$=$\{h_1, h_2, ..., h_n\}$, where $X \in \mathbb{R}^{d\times n}$, $H \in \mathbb{R}^{d\times n}$. $d$ and $n$ are the dimension of hidden layers and length of sequence. To implement our non-symmetrical window strategy, we split the whole window $D$ into $D_{Left}$ and $D_{Right}$, which represent the windows on both sides of headword $x_j$. In most case, $D_{Left}$ is different from $D_{Right}$, and the extra distribution can be defined as Eq. (4)

$$\begin{cases} \theta_{i,j} = \left|\frac{Cov|(P_i,j)|}{\sigma_i \cdot \sigma_j}\right| - 0.5 \\ GL_{i,j} = max\left(0, \theta_{i,j}\right) \frac{2\theta_{i,j}(j-P_i)^2}{D^2_{Left}} \\ GR_{i,j} = max\left(0, \theta_{i,j}\right) \frac{2\theta_{i,j}(j-P_i)^2}{D^2_{Right}}, \end{cases} \tag{4}$$

where $P_i$ is the position of predicted words adjacent to $x_i$. Since the prediction of each headword depends on its corresponding scalar $p_i$, it can be calculated by Eq. (5). $\theta_{i,j} \in [-0.5, 0.5]$ is a coefficient that controls the extent of

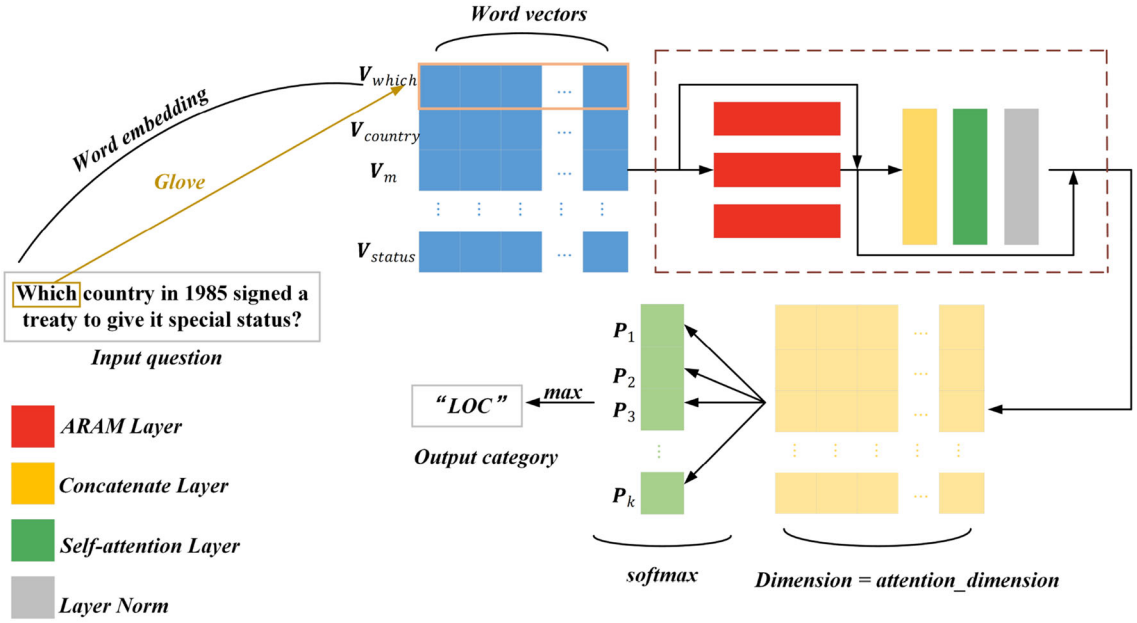


**Fig. 1** The whole architecture of the proposed POS-ARAN model

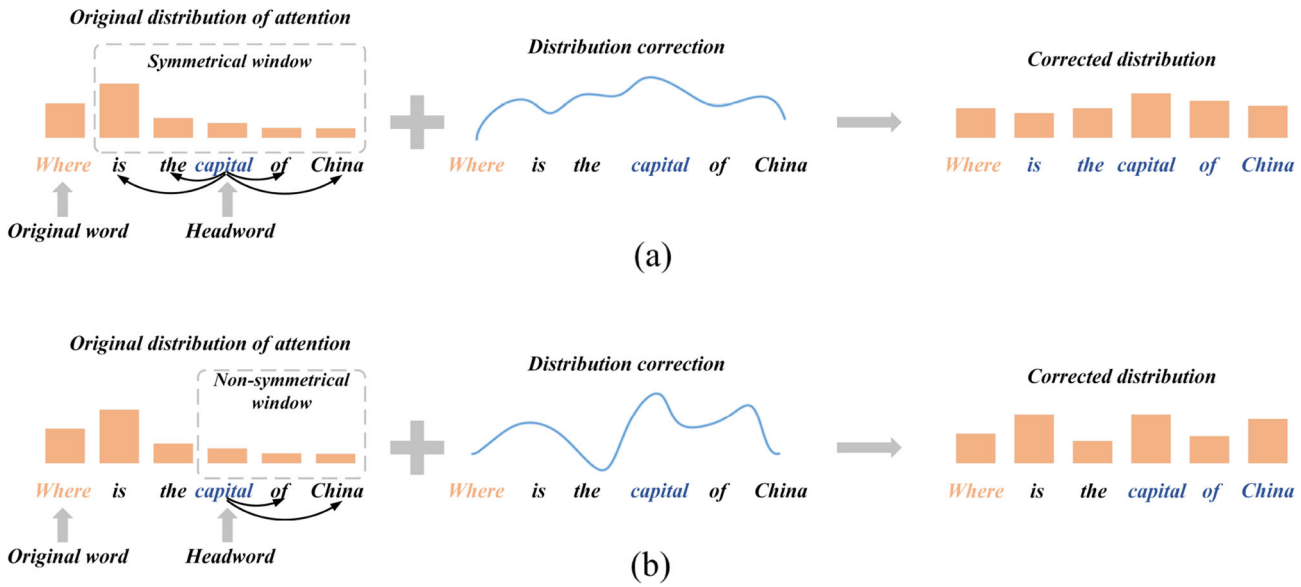


**Fig. 2** Distribution correction of symmetrical window and non-symmetrical window

correction. When $\theta_{i,j} > 0$, the attention of the corresponding position should be correct. $GL_{i,j}$ and $GR_{i,j}$ denote the Gaussian distribution based on the left window and right window of the headword. Traditionally, $D$ is set as a constant [44], which means that we can just improve the adjacent relation in a fixed range around the headword. However, this would be hindered by two problems. First, the value of $D$ is hard to determine. If $D$ is too big, the improvement of the adjacent relation is pointless, and if $D$ is too small, some important words may be overlooked. Second, the window size of headwords differs from each other. In other words, this value should be determined dynamically. Therefore, ARAM itself learns a transition matrix $W_d$ to calculate $D$ for each headword dynamically, as shown in Eq. (6)

$$P_i = \left\lceil R / \left[ 1 + \exp\left( -U_p^T \tanh\left( W_i x_i \right) \right) \right] \right\rceil \tag{5}$$

$$D_{\text{Left/Right}} = \left\lceil \lambda R / \left[ 1 + \exp\left( -U_d^T \tanh\left( W_d x_i \right) \right) \right] \right\rceil, \tag{6}$$

where $U_d^T$ is a set of linear projection vector. $W_d \in R^{d\times d}$ denotes the transition matrix learned dynamically during training. $R$ is a scaled factor that rescale the range of window, and $\lambda$ is a fine-tuned weight. Then, the process of the revised self-attention in our POS-ARAN model can be formalized as Eq. (9)

$$e_{ij} = \left(x_i \cdot W^Q\right) \cdot \left(x_j \cdot W^K + r_{ij}^K\right) / \sqrt{d} \tag{7}$$

$$a_{ij} = \frac{\exp\left(e_{ij} + GL_{ij} + GR_{ij}\right)}{\sum_{k=1}^{n} \exp\left(e_{ik} + GL_{ik} + GR_{ik}\right)} \tag{8}$$

$$h_i = \sum_{j=1}^{n} a_{ij} \cdot \left(x_j \cdot W^V + r_{ij}^V\right), \tag{9}$$

where $W^Q$, $W^K$,$W^V \in \mathbb{R}^{d_x \times d_h}$ are the transition matrices of vector Query, Key, and Value. In Eq. (8), $GL_{i,j}$ and $GR_{i,j}$ are added to the original distribution linearly. Equation (7) applies the scaled dot product of self-attention model to define the scoring function of $e_{ij}$. This method adds a scaled factor on the basis of the traditional dot product, which can avoid the low learning efficiency caused by small gradient. To scale the result to an appropriate range, we set the scaled factor as $1/\sqrt{d}$. Moreover, during training process, we assume that when the distance between two elements in the same sequence exceeds a certain threshold value $k$, then the information of these two elements is less important. In Eqs. (7) and (9), we add a bias $r_{ij}$ to describe the adjacent relation between $x_i$ and $x_j$. $r_{ij}$ is transferred from $PPE$, which is a novel positional encoding method introduced in the next subsection. Thereby, the calculation of $r_{ij}^V$ and $r_{ij}^K$ can essentially be attributed to training two relative position sequences $W^V$ and $W^K$

$$\begin{cases} W^V = \left\{w_{-k}^V, \cdots, w_k^V\right\}, w_i^V \in R^{d_h}, i \in [-k, k] \\ W^K = \left\{w_{-k}^K, \cdots, w_k^K\right\}, w_i^K \in R^{d_h}, i \in [-k, k] \end{cases} \tag{10}$$

$$\begin{cases} r_{ij}^V = PPE \cdot W_{max(-k,min(j-i,k))}^V \\ r_{ij}^K = PPE \cdot W_{max(-k,min(j-i,k))}^K \end{cases}. \tag{11}$$

As illustrated in Fig. 3, the ARAM module consists of three "Adjacent Relation Promotion" blocks, one "Concatenate" layer, one "Multi-Head Attention" layer. and one "Add & Norm" layer, which is the structure verified by experiments. First, we pass the input word vector matrix through "Adjacent Relation Promotion" blocks simultaneously to learn the features of sequence in different subspaces. Then, we concatenate three outputs and use the self-attention layer to capture the dependency of context. Finally, we use the layer normalization [45] which could perform well in the sequence problems of NLP.

In addition, we try to cascade the ARAM module $N$ times to make the network deeper for better performance. However, deeper networks are also difficult to train due to the problem of vanishing gradient. To avoid this problem, we apply the residual connection which is denoted by the blue arrow in Fig 3. The experimental results shown in Section "Experiments" have proved its effectiveness.

## POS-aware embedding layer

Inspired by [46], we find that the part-of-speech (POS) of word is also an available feature that can guide the original words to locate the appropriate headwords. As shown in Fig. 4, "what" is much more related to "abbreviation" and "AIDs", less related to "stand" and "for" and barely related to "does" and "the". In this case, we can find that the words with tags "*WP*", "*NN*", and "*NNP*" have stronger relation. This is not a particular case. By analyzing plenty of samples, a strong semantic relationship can be inferred via the words with a given POS (e.g., pronoun and noun in a question usually have a strong relevance). Generally, the conventional word embedding is to transform a word into the vector representation, which does not contain POS information. Accordingly, we introduce a POS-aware embedding layer as an alternative to the word embedding layer. As depicted in Fig. 5, our model employs GloVe [47] to implement the word embedding process, which could generate a matrix $W^R$ with a shape of $m \times n$. $m$ and $n$ denote the dimension of word vector and sequence length, respectively. Similar to the self-attention, given the POS tag of the inputting sentence $Y = \{y_1, y_2, ..., y_n\} \in \mathbb{R}^{n\times 1}$ and a learnable parameter $W^S \in \mathbb{R}^{n\times m}$, the POS vector $S \in \mathbb{R}^{m\times 1}$ can be defined as

$$S = \left(W^S\right)^T Y. \tag{12}$$

Then, to integrate the original word representation with the POS information, the POS vector $S \in \mathbb{R}^{m\times 1}$ is extended to a matrix $S' \in \mathbb{R}^{m\times n}$. The final word representation $\left(W^R\right)'$ can be illustrated as

$$W^{R'} = W^R \cdot S', \tag{13}$$

where $\cdot$ denotes dot product and the final word representation consists of essential word embedding and the POS of each word.

Different from other POS-based solutions [48, 49], we combine the POS embedding with positional encoding. In self-attention network, positional encoding is indispensable, because self-attention network processes all words in sentence simultaneously, rather than one by one as in RNNs.

In other words, previous self-attention networks did not utilize position information. Vaswani et al. [50] proposed

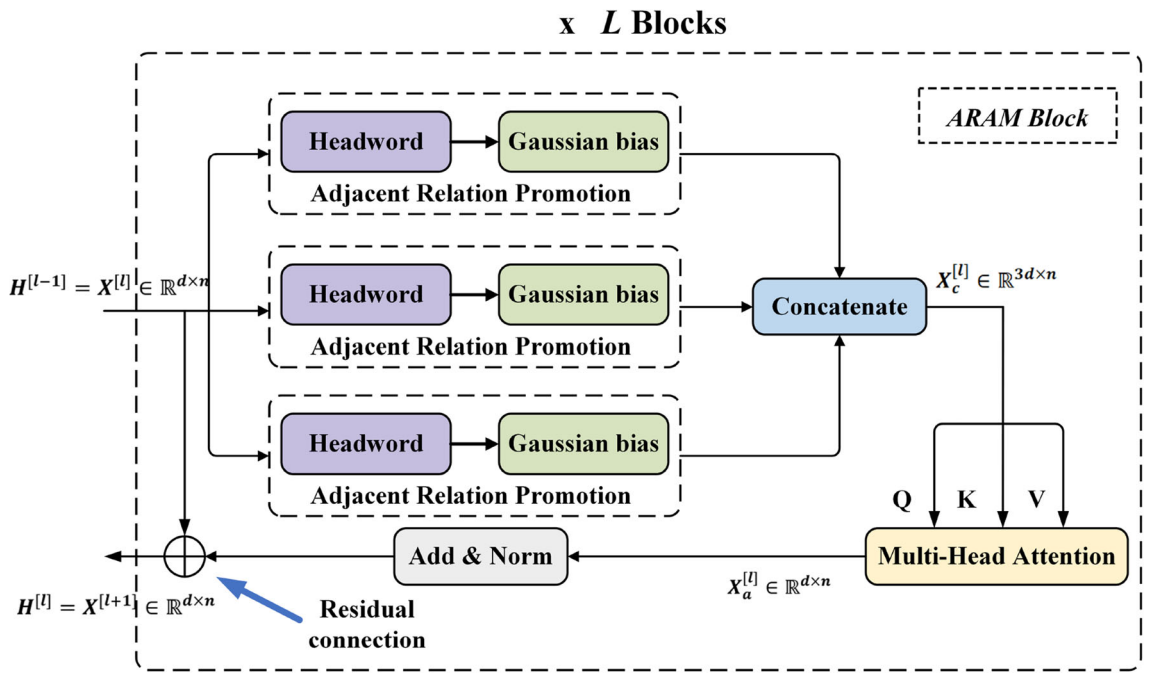


**Fig. 3** The structure of ARAM module. "Adjacent Relation Promotion" improves the adjacent relation of words, which utilizes the Gaussian bias to correct the distribution of self-attention. "Concatenate" concatenates the vector matrices from the previous layer. "Multi-Head Attention" is the self-attention layer. "Add & Norm" represents the layer normalization

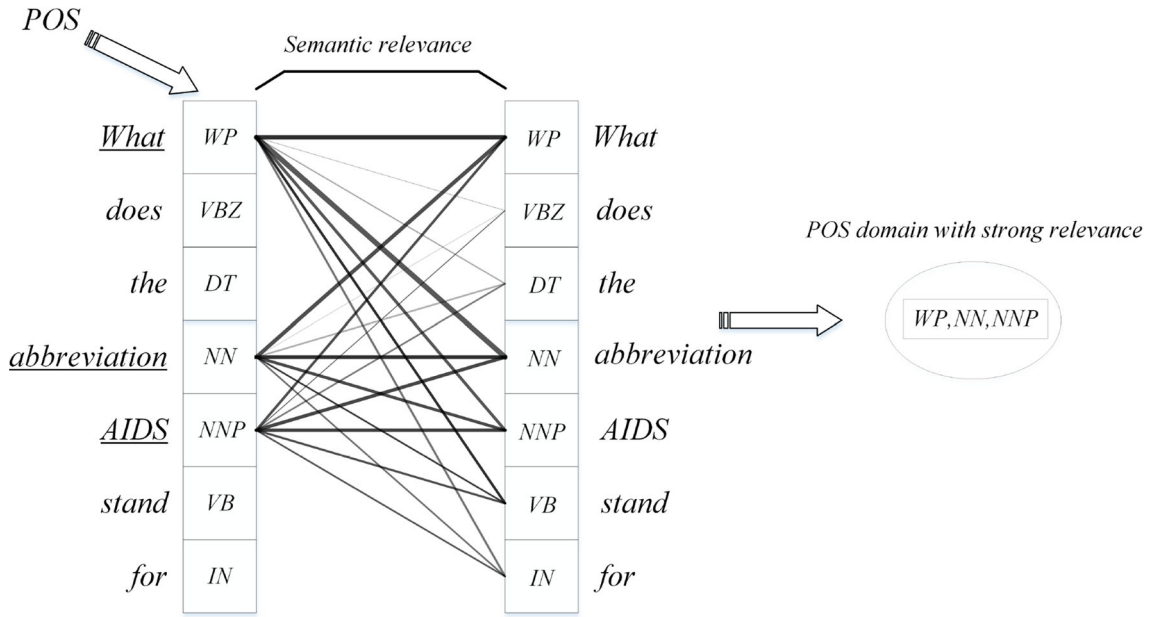


**Fig. 4** The relation between semantic relevance and POS. The abbreviations for the part-of-speech tags correspond as follows: $WP$: pronoun, $VB(VBZ)$: verb, $DT$: determiner, $NN(NNP)$: noun, $IN$: preposition

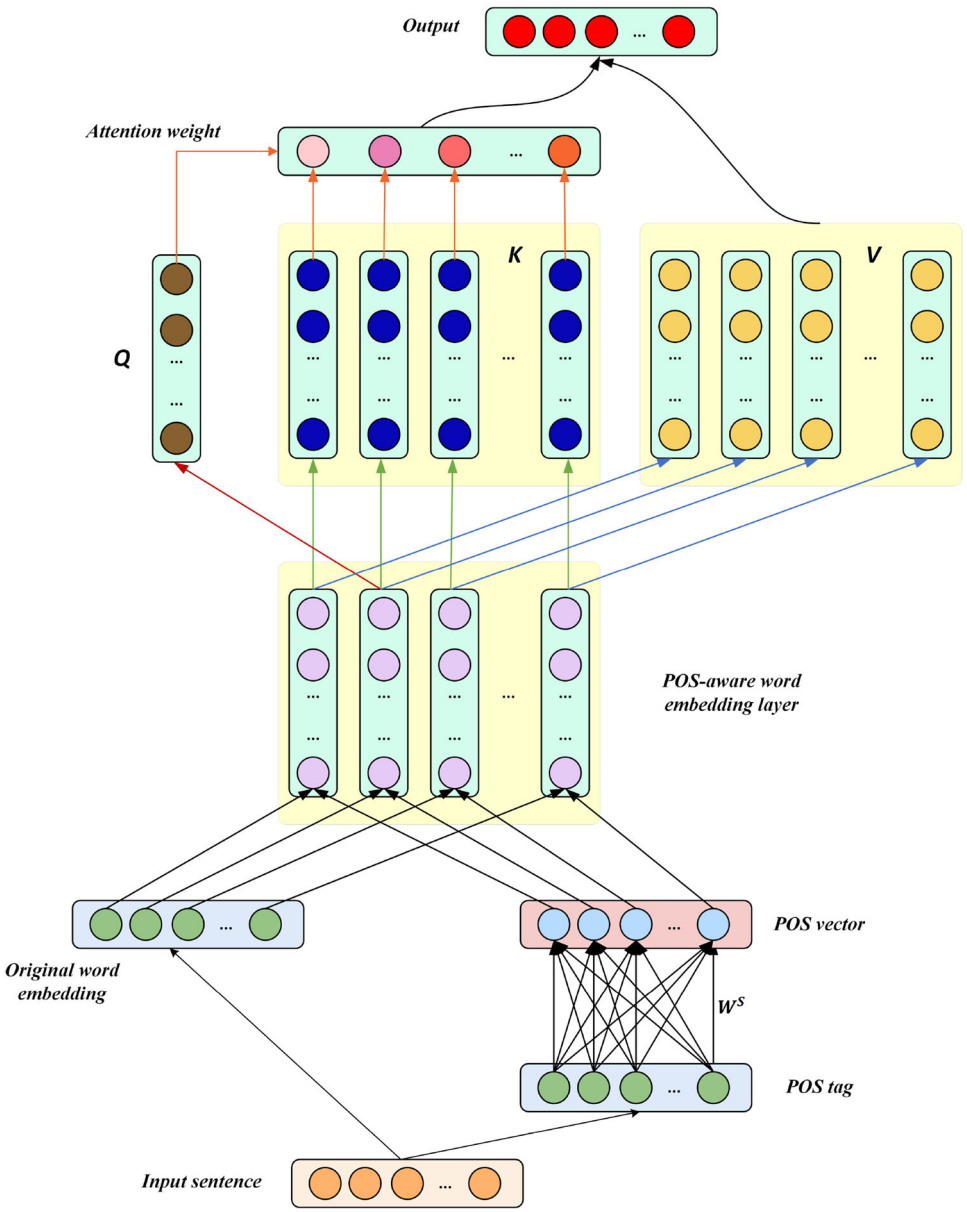


**Fig. 5** The detailed process of revised self-attention

positional encoding to embed positional information into self-attention network. Their method is defined as Eqs. (14) and (15)

$$PE_{(p,2i)} = \sin\left(p/10000^{2i/d_{model}}\right) \tag{14}$$

$$PE_{(p,2i+1)} = \cos\left(p/10000^{2i/d_{model}}\right), \tag{15}$$

where $p$ denotes the position of word in sentence, $i$ represents the $i$th element of the positional vector, and $d_{model}$ is the dimension of word vector. We consider that the effect of the same POS on the attention distribution may vary from one position to another. Also, positional encoding does not require any training parameters and the output can be directly normalized to [-1,1] by trigonometric functions. Thence, we combine POS embedding with positional encoding. The detailed process is formalized in Eq. (16)

$$PPE_{(POS,p,t)} = \begin{Bmatrix} sin\left(p \cdot e^{-4tlog\frac{10}{d_{model}}+POS\cdot t}\right), t = 0, 2, 4, \ldots \\ cos\left(p \cdot e^{-4(t-1)log\frac{10}{d_{model}}+POS\cdot t}\right), t = 1, 3, 5, \ldots \end{Bmatrix}, \tag{16}$$

where $POS$ is the POS feature of words. $PPE$ denotes the POS-aware embedding with positional encoding, which is a matrix with a shape of $m \times n$. $m$ and $n$ denote the dimension of word vector and sequence length, respectively. $PPE_{(p+k)}$ can be represented by the linear function of $PPE_{(}p)$ accord-

**Table 1** Experimental environment

| Name | Setting |
|---|---|
| CPU | E5-2630@2.40GHz |
| RAM | 64GB |
| GPU | GTX2080Ti |
| OS | Ubuntu 16.04 |
| Programming language | Python3.6 |
| IDE | Pycharm |
| Framework | Keras 2.2.4 |

**Table 2** Coarse-grained data distribution

| Category | Quantity |
|---|---|
| ABBR | 241 |
| NUM | 2480 |
| ENTY | 3626 |
| HUM | 3422 |
| LOC | 2375 |
| DESC | 3303 |

ing to Eqs. (17) and (18). It means that we can judge the positional relation between two words by the value of $k$

$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta \tag{17}$$

$$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta. \tag{18}$$

As the POS features are embedded in the PPE, its influence can vary with relative position. Additionally, the computation process nearly has no training parameters, which will not incur much extra cost.

Then, we split the word representation into $h$ heads and utilize several transition matrices to calculate the $Q/K/V$ matrices for attention $Z_i$. $Z_i$ is calculated by Eq. (19)

$$Z_i = softmax\left(\frac{Q_i^T K_i}{\sqrt{d_k}}\right) V, i \in [0, h-1]. \tag{19}$$

Finally, all $Z_i$ are concatenated together and multiplied with a transition matrix to calculate the final $Z$. $Z$ has the same dimension with the initial word representation.

## Experiments

### Experimental environment

The experimental environments are listed in Table 1. We choose Pycharm and Keras to implement our experiments on a Linux physical machine with 64GB RAM, E5-2630@2.40GHz CPU and GTX2080Ti GPU.

### Datasets

We have evaluated our POS-ARAN on two different question classification datasets: (i) Experimental Data for Question Classification (EDQC) and (ii) Yahoo! Answers Comprehensive Questions and Answers 1.0.

EDQC dataset is a public question classification dataset [51], which can be obtained from https://cogcomp.seas.upenn.edu/Data/QA/QC/. The dataset contains 15,447 questions in total and all questions are labeled according to coarse grain and fine grain, respectively. On the coarse-grained level, the intent of all questions can be divided into 6 categories: "ENTY" (entity), "LOC" (location), "NUM" (number), "DESC" (description), "HUM" (human), and "ABBR" (abbreviation). Moreover, all questions are also labeled with 47 fine-grained categories. The category information and data distribution are listed in Tables 2 and 3.

The Yahoo! Answers topic classification dataset is constructed using 10 largest main categories, published by Cornell University, including Society & Culture, Science & Mathematics, Health, Education & Reference, Computers & Internet, Sports, Business & Finance, Entertainment & Music, Family & Relationships and Politics & Government. Each class is comprised of 140,000 training samples and 5000 testing samples. The dataset is a corpus of answers by the end of October 25, 2007 at Yahoo, which includes all questions and their corresponding answers. In the question classification experiments, we only used the question and the main category information.

### Evaluation metrics

Multiple performance and evaluation criteria are used to evaluate the performance of proposed model. Following prior work, we adapt Accuracy, Precision($P$),Recall($R$) and F1 score. The formulas are stated in Eqs. (20)–(22)

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{20}$$

$$\mathrm{F1} = \frac{2\times P\times R}{P+R} \tag{21}$$

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+TN}, \tag{22}$$

where $TP$ represents the number of samples correctly predicted as positive class, $FP$ represents the number of samples incorrectly predicted as positive class, $TN$ is the number of samples correctly predicted as negative class, and $FN$ is the number of samples incorrectly predicted as negative class. The same is true for multiple classifications, as long as all other categories that do not belong to the current category are considered as negative cases. Higher values denote better performance for all metrics.

**Table 3** Fine-grained data distribution

| Category | Quantity | Category | Quantity | Category | Quantity | Category | Quantity |
|---|---|---|---|---|---|---|---|
| Speed | 36 | Period | 197 | Food | 266 | Word | 71 |
| Currency | 5 | Dist | 86 | Date | 650 | Gr | 529 |
| Temp | 15 | Letter | 30 | Symbol | 31 | Substance | 124 |
| Code | 22 | Exp | 195 | Count | 985 | Def | 1220 |
| Other | 2088 | Body | 54 | Desc | 911 | Abb | 46 |
| Sport | 165 | Religion | 9 | Country | 425 | State | 201 |
| Techmeth | 111 | Ord | 15 | Mount | 67 | Product | 130 |
| Lang | 45 | Dismed | 291 | Perc | 82 | Weight | 23 |
| City | 376 | Veh | 68 | Plant | 38 | Instru | 37 |
| Title | 67 | Animal | 365 | Volsize | 32 | Cremat | 595 |
| Termeq | 271 | Manner | 766 | Event | 173 | Reason | 543 |
| Color | 119 | Ind | 2689 | Money | 183 | | |

**Table 4** Hyperparameter configuration

| Hyperparameter | Setting |
|---|---|
| Input Layer | Seq_length=50 |
| Embedding Layer | Dimension=100 |
| POS-ARAN Layer I | Heads=4, dimension=64 |
| POS-ARAN Layer II | Heads=4, dimension=32 |
| Dropout | 0.5 |
| Dense Layer | Dimension=class_numbers |
| Optimizer | ADAM |
| Loss Function | Categorical cross-entropy |
| Batch_size | 64 |
| Validation_split | 0.3 |
| Learning_rate | 0.001 |
| Epoch | 200 |

## Parameter configuration

The hyperparameters in POS-ARAN are listed in Table 4. We assume that the length of input sequence does not exceed 50. If the sequence length is shorter than 50, we will pad it to 50. After word embedding, the shape of input sequence will be $n \times 100$, where $n$ is the sequence length. To avoid overfitting, we set the dropout rate as 0.5. The final layer of POS-ARAN is a dense layer, whose dimension is consistent with the number of categories. We select categorical cross-entropy as our loss function, which is formulated in Eq. (20)

$$c = -\frac{1}{n}\sum_{x} y \ln a + (1-y)\ln(1-a). \tag{23}$$

To update parameters effectively, we use ADAM algorithm to optimize POS-ARAN model. As for the hyperparameters of ADAM, we set $\beta_1 = 0.9$, $\beta_1 = 0.999$, $\epsilon = 10^{-8}$. 70% of the data is divided into training set and the remaining 30% is divided into validation set. We set the learning rate as 0.001 and train the model for 200 epochs. After observation, the performance of POS-ARAN model keeps stable after 50 epochs.

## Experimental results and analysis

In this section, a detailed experimental result and analysis are presented. As depicted in Fig. 6, the accuracy of coarse-grained and fine-grained classification task is 95.59% and 92.91%. It is obvious that POS-ARAN performs better on coarse-grained classification tasks. The main reason is that fine-grained categories are much more than coarse-grained categories, so that the number of samples in every fine-grained category is not enough. As a result, the difficulty of fine-grained classification is greatly increased, which makes the difference of validation accuracy reasonable. Correspondingly, the training loss of coarse-grained classification task is lower than that of fine-grained classification task. In addition, compared with the result of fine-grained classification task, the training accuracy of coarse-grained classification task converges more quickly. The experimental results show that POS-ARAN can successfully converge in 200 epochs and achieves satisfactory performance both on coarse-grained classification task and fine-grained classification task.

To understand how the ARAM module assigns its attention, we visualize the weights distribution of attention in Fig. 7. As presented in Fig. 7, an attention weight matrix is constructed to demonstrate the correlation between any two words in the sentence. The darker the elements in the matrix are, the more contributions the relative words have when calculating the attention weights. In this way, neighboring words can have the same contribution as headwords. The mechanism is beneficial to capture the local semantic information

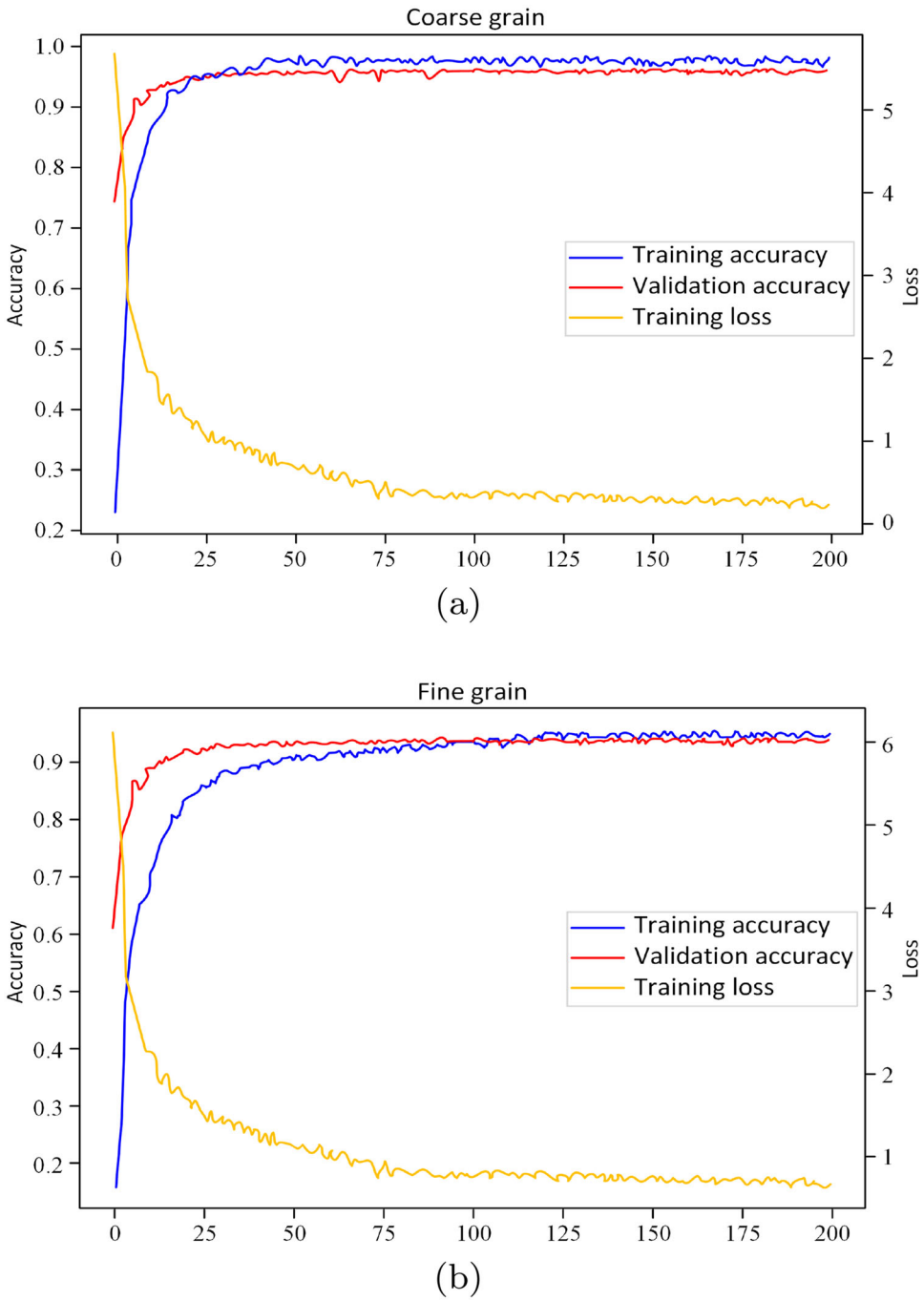


**Fig. 6** Visualization of the training accuracy

in sentences, which contributes to correctly judge the intent of questions.

As shown in Fig. 8, the visualization results of ARAM are composed of Query $q$ Key $k$ and element-wise $q \times k$, which gives us a clear picture of the intermediate results of queries and keys. It helps to demonstrate the robustness of the POS-ARAN model in more detail. According to the result output by Softmax layer, "what" is more related to "term", and meanwhile keeps a certain relevance with "is", "the", "young", and "fox". It is believed that our POS-ARAN model can capture the short-term and long-term dependency simultaneously by capturing the adjacent relation.

## Comparison results and ablation studies

In this section, we compare our POS-ARAN model with the traditional machine learning methods and other mainstream deep learning networks on EDQC and Yahoo! Answers. The comparison results are shown in Table 5. In our experiments, SNoW [53] is an improved hierarchical classification model.

**Table 5** Comparison of the proposed POS-ARAN against State-of-the-Art Benchmark Algorithms

| Dataset | Model | Metrics (%) | | | |
|---|---|---|---|---|---|
| | | Acc | F1 | P | R |
| EDQC (Coarse grain) | SNoW | 89.3 | 88.76 | 90.32 | 87.26 |
| | Maximum Entropy | 89.8 | 89.61 | 88.78 | 90.46 |
| | Linear_SVM | 91.2 | 89.92 | 91.34 | 88.54 |
| | GRU | 93.54 | 92.72 | 92.55 | 92.89 |
| | TextCNN | 91.56 | 88.35 | 91.65 | 85.27 |
| | TextRNN | 94.79 | 93.66 | 93.68 | 93.99 |
| | CNN+LSTM | 94.2 | 91.04 | 90.26 | 91.83 |
| | Bi-LSTM | 95.08 | 95.46 | 96.06 | 94.86 |
| | Word2vec+CNN | 92.75 | 89.67 | 92.23 | 87.31 |
| | Word2vec+RNN | 94.81 | 93.83 | 93.91 | 93.36 |
| | MGF | 91.97 | 88.24 | 87.74 | 88.69 |
| | BERT | 95.25 | 95.02 | 94.36 | 95.68 |
| | ALBERT | 95.48 | 95.94 | 96.8 | 95.1 |
| | **POS-ARAN** | **95.59** | **96.45** | **96.62** | **96.28** |
| EDQC (Fine grain) | SNoW | 84.2 | 84.19 | 85.23 | 83.18 |
| | Maximum Entropy | 85.4 | 85.3 | 84.24 | 86.38 |
| | Linear_SVM | 84.23 | 83.54 | 85.37 | 81.78 |
| | GRU | 87.51 | 86.68 | 86.49 | 86.87 |
| | TextCNN | 85.04 | 83.91 | 85.24 | 82.63 |
| | TextRNN | 89.66 | 88.83 | 88.68 | 88.98 |
| | CNN+LSTM | 91.37 | 89.96 | 89.35 | 90.57 |
| | Bi-LSTM | 91.97 | 91.48 | 92.95 | 90.06 |
| | Word2vec+CNN | 87.42 | 85.65 | 87.23 | 84.81 |
| | Word2vec+RNN | 91.31 | 89.83 | 89.71 | 90.27 |
| | MGF | 89.97 | 87.36 | 87.64 | 86.49 |
| | BERT | 92.46 | 92.14 | 91.44 | 92.86 |
| | ALBERT | 92.76 | 92.97 | 93.78 | 92.18 |
| | **POS-ARAN** | **92.91** | **92.42** | **91.76** | **93.08** |
| Yahoo Answers! | SNow | 57.35 | 57.33 | 58.39 | 56.31 |
| | Maximum Entropy | 58.7 | 58.79 | 57.77 | 59.85 |
| | Linear_SVM | 59.75 | 58.87 | 59.93 | 57.85 |
| | GRU | 60.55 | 59.79 | 59.65 | 59.93 |
| | TextCNN | 58.64 | 57.59 | 58.86 | 56.38 |
| | TextRNN | 60.58 | 59.76 | 59.64 | 59.88 |
| | CNN+LSTM | 61.78 | 60.88 | 60.48 | 61.26 |
| | Bi-LSTM | 62.06 | 62.9 | 63.95 | 61.89 |
| | Word2vec+CNN | 60.31 | 60.55 | 60.53 | 61.18 |
| | Word2vec+RNN | 62.71 | 62.83 | 62.71 | 62.27 |
| | MGF | 61.97 | 61.33 | 60.64 | 61.49 |
| | BERT | 63.46 | 63.71 | 62.44 | 63.96 |
| | ALBERT | 64.28 | 64.65 | 64.92 | 64.38 |
| | **POS-ARAN** | **65.73** | **65.54** | **65.78** | **65.31** |

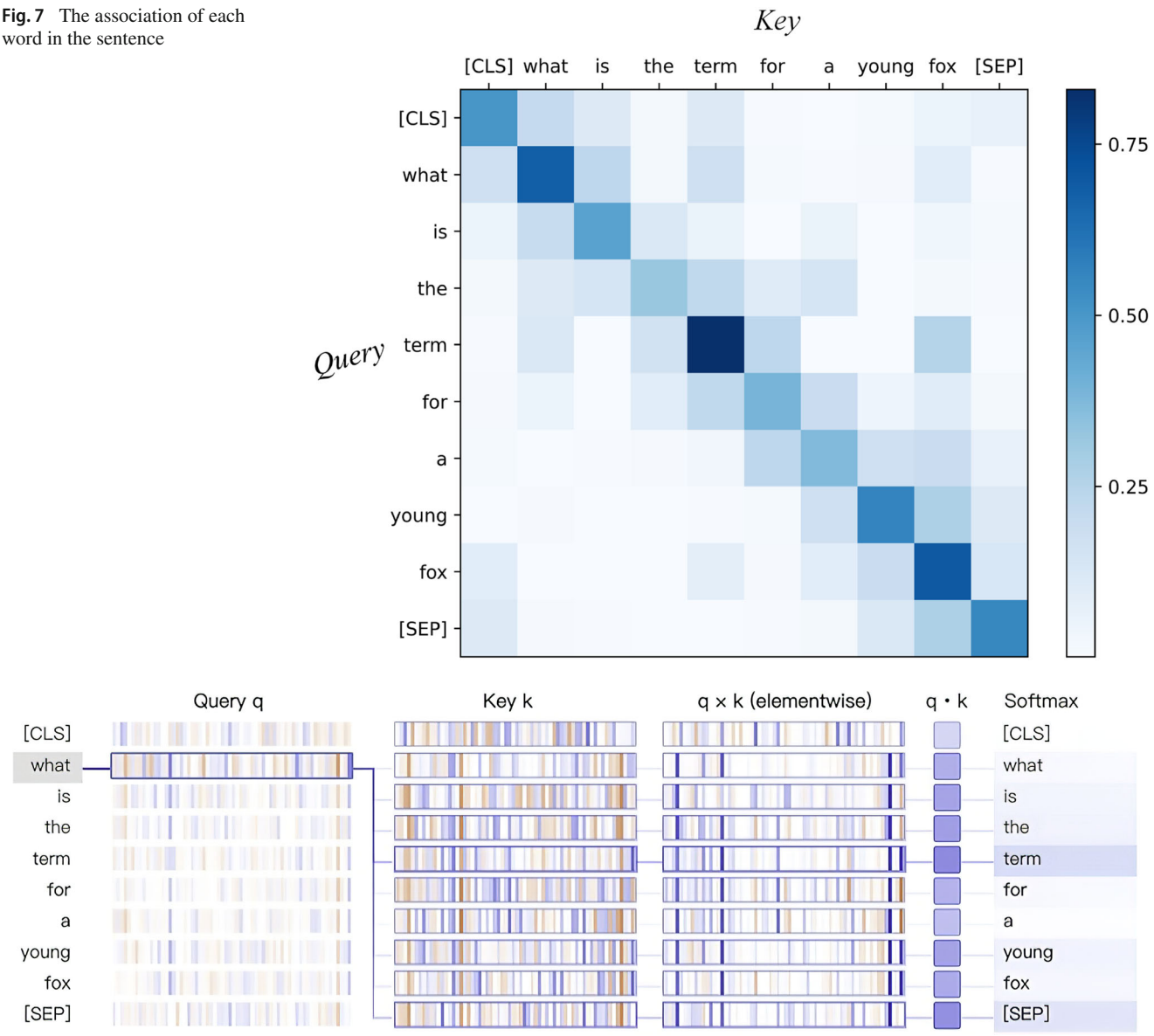


**Fig. 7** The association of each word in the sentence

**Fig. 8** The visualized result of the improved self-attention. "[CLS]" and "[SEP]" are two delimiters mentioned in [52] and have no semantic information

Maximum entropy model is the one proposed by Kocik [33] et al. SVM is the support vector machine with linear kernel proposed by Huang [31] et al. TextCNN [37] and TextRNN [54] are the common models applied to sequence problems. GRU [55] and Bi-LSTM [56] are two special variants of RNN which perform well in NLP task. Word2vec pre-trained word embeddings [36] are introduced on deep learning architectures for experiments including CNN and RNN. Bidirectional Encoder Representations from Transformers (BERT) [57] achieve state-of-the-art performance, since it can provide better representation through capturing bi-directional context using Transformers. ALBERT [58] is an extension study on BERT, which improves parameter-efficiency of BERT by incorporating two parameter reduction techniques. A multi-granularity fusion neural network (MGF) [59] is used for comparison, which has recently achieved optimal results in medical question classification. More intuitive results can be observed according to Fig. 9. Especially, since POS-ARAN model is essentially a deep learning model, we also compare the accuracy trend of our POS-ARAN with other deep learning methods, as shown in Fig. 10.

According to Fig. 9 and Table 5, it is obvious that deep learning methods are generally better than the traditional machine learning methods. Specifically, our POS-ARAN model performs well no matter in coarse-grained task or fine-grained. Although the accuracy of GRU and Bi-LSTM

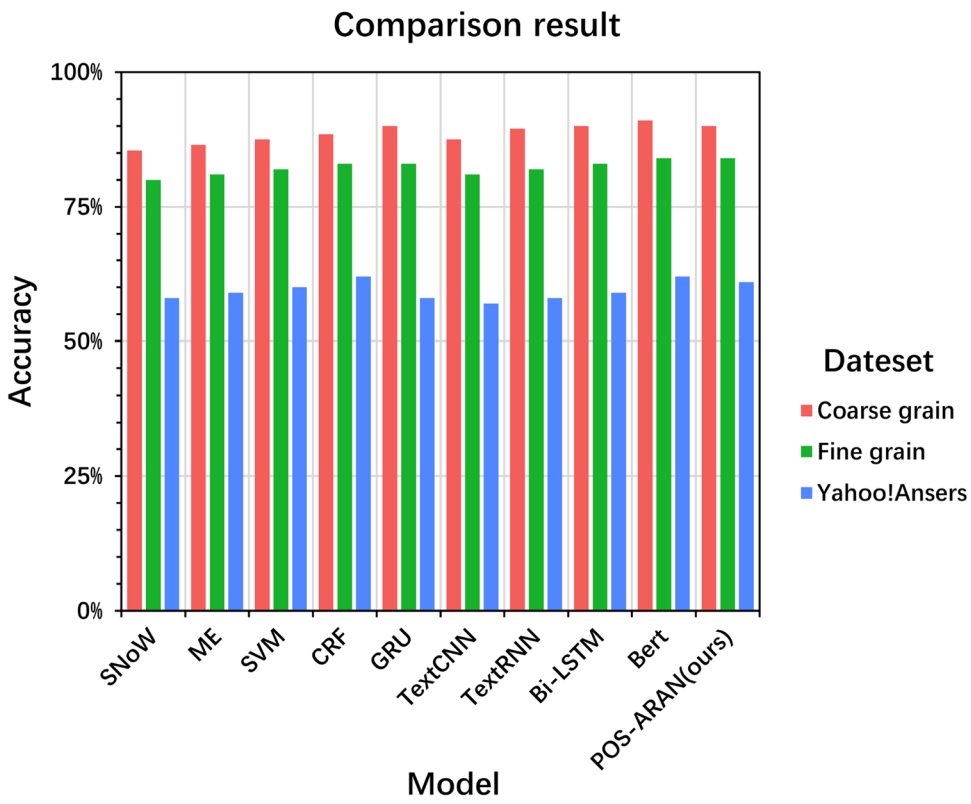


**Fig. 9** The comparison results of various models

**Table 6** Results of coarse-grained category in EDQC of POS-ARAN

| Coarse-grained category | | | | | | |
|---|---|---|---|---|---|---|
| | ABBR | DESC | ENTY | HUM | LOC | NUM |
| Precision | 0.974 | 0.942 | 0.942 | 0.974 | 0.959 | 0.988 |
| Recall | 0.755 | 0.965 | 0.945 | 0.966 | 0.966 | 0.97 |
| F1 | 0.851 | 0.954 | 0.944 | 0.97 | 0.97 | 0.979 |

is similar in accuracy to POS-ARAN on coarse-grained task, POS-ARAN still outperforms them on fine-grained task. Compared to the Bert, the current state-of-the-art approach for NLP tasks, our POS-ARAN still achieves a competitive performance. As shown in Tables 6 and 7, our proposed model POS-ARAN can classify well the NUM category in EDQC and Sports category in Yahoo! Answers. Meanwhile, we find that the recall of ABBR category is lower than other categories, which may be related to the insufficient sample size.

Since the architecture of GRU and Bi-LSTM is essentially RNN, it means that they are trained serially. As shown in Fig. 10, we can observe that RNN-based models take a long time to train and converge more slowly than POS-ARAN. Figure 11 illustrates the training procedure for various models and demonstrates the variance in training duration. As can be seen from Fig. 11, TextCNN is the fastest, followed by POS-ARAN. GRU and TextRNN take nearly three times longer than POS-ARAN and yet Bi-LSTM is the worst, with a time expenditure of around six times that of POS-ARAN. Thus, although the accuracy of GRU and Bi-LSTM is on par with our POS-ARAN model, POS-ARAN still has a significant advantage in terms of training speed. In contrast, TextCNN, although faster than POS-ARAN, is far less accurate. In brief, POS-ARAN achieves optimal results compared to other methods across multiple evaluation metrics.

Moreover, to obtain the best performance, we try several different combinations of parameters and compare them on EDQC, as shown in Table 8, and Figs. 12 and 13. The result shows that the performance of POS-ARAN model increases continuously as the increasing of ARAM layers, attention heads, and attention dimension. In the meantime, we also need to take the number of parameters into consideration. Blindly increasing parameters will lead to excessive computational expenditure. Finally, we obtain the best architecture of POS-ARAN model in our experiments, which has 2 ARAM layers, 4 attention heads, and attention dimension of 64.

## Discussion

According to the final qualitative result, we find that almost all coarse-grained categories are correctly classified; however, some questions are misclassified in fine-grained tasks.

**Table 7** Results of POS-ARAN on Yahoo! Answers

| Yahoo! Answers | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bussiness | Computers | Education | Entertainment | Family | Health | Politics | Science | Society | Sports |
| Precision | 0.441 | 0.776 | 0.447 | 0.663 | 0.599 | 0.663 | 0.679 | 0.653 | 0.598 | 0.824 |
| Recall | 0.485 | 0.8 | 0.475 | 0.591 | 0.71 | 0.702 | 0.692 | 0.629 | 0.448 | 0.774 |
| F1 | 0.462 | 0.787 | 0.46 | 0.625 | 0.65 | 0.682 | 0.685 | 0.641 | 0.513 | 0.798 |

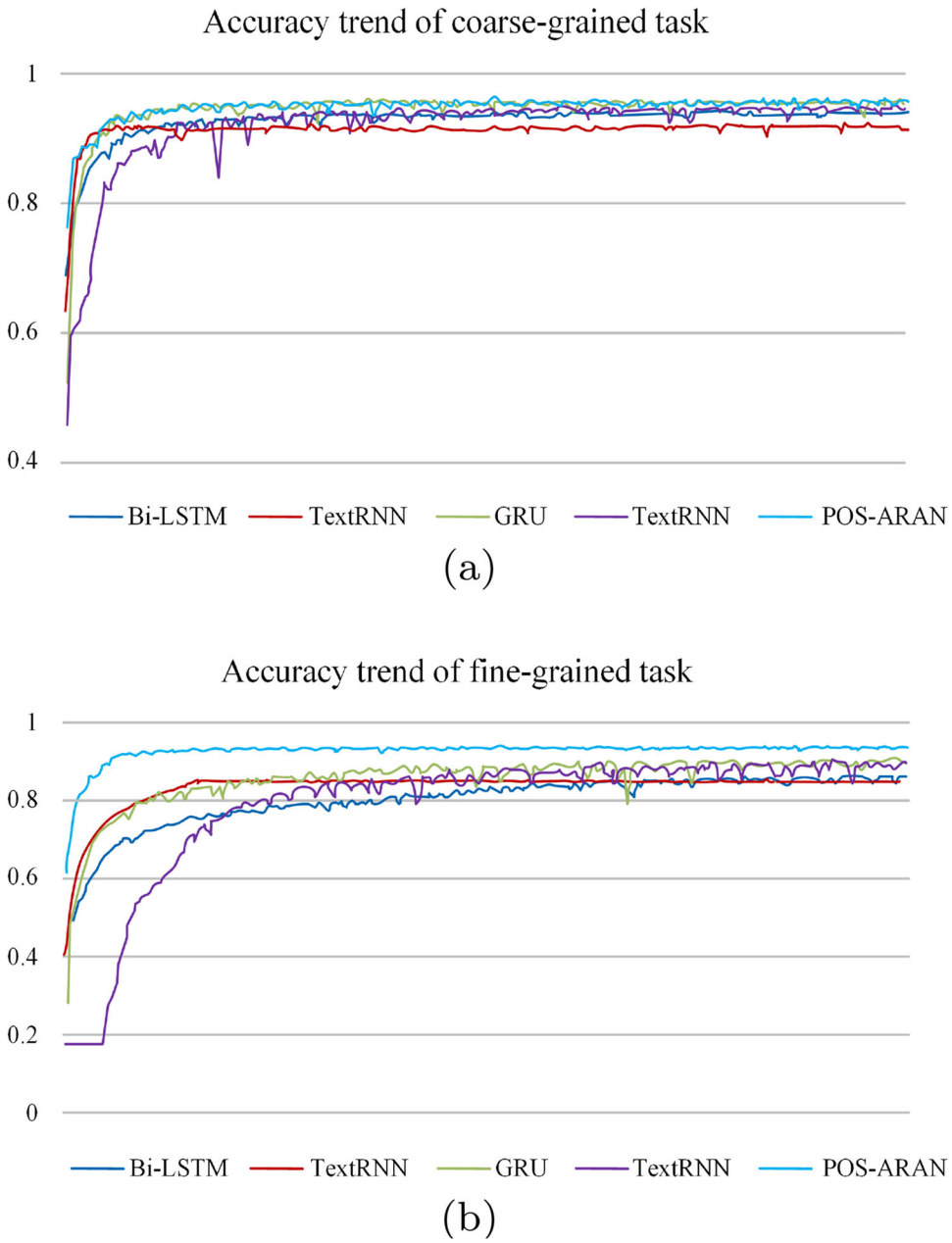


**Fig. 10** The coarse-grained and fine-grained accuracy trend

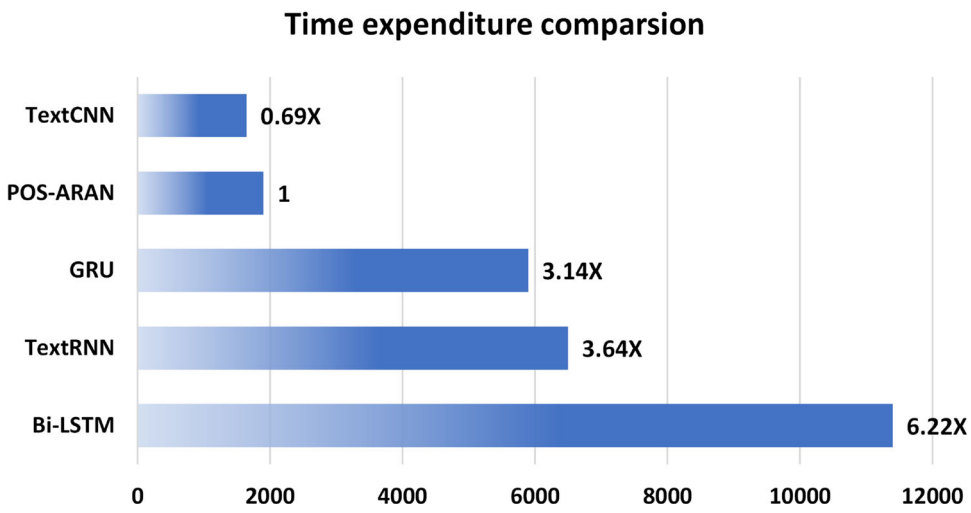


**Fig. 11** The difference of various models' time expenditure

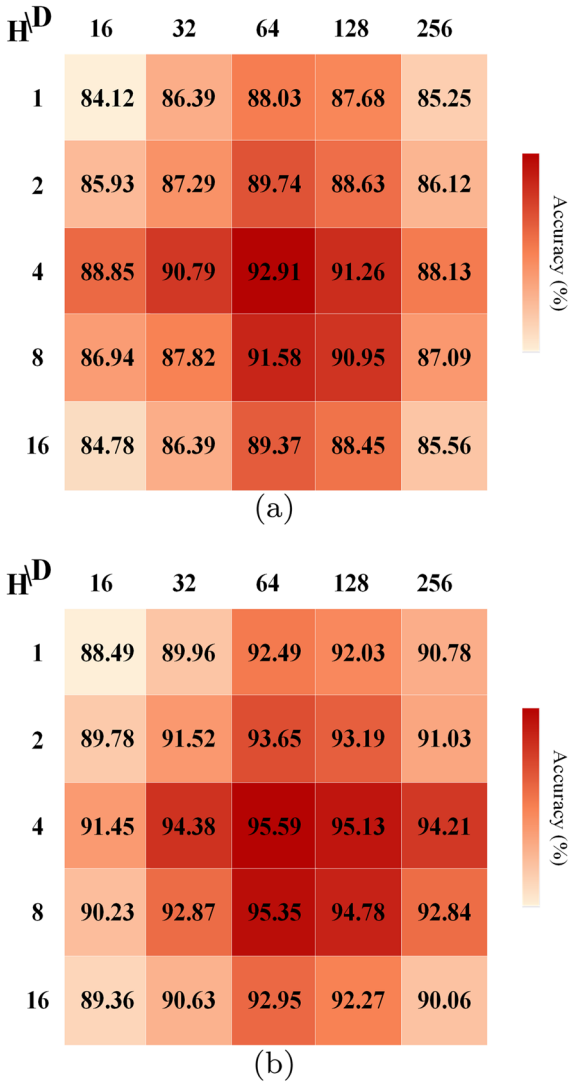


**Fig. 12** Heatmap of accuracy (%) on EDQC of different configurations

**Table 8** Accuracy of POS-ARAN model with different configurations."ARAM layers" represents the number of layers of ARAM. "Attention heads" (Attr-H) is the number of head in self-attention mechanism. "Attention dimension" (Attr-D) denotes the dimension of the hidden layer in attention network

| Model | Configuration | | | Accuracy | |
|---|---|---|---|---|---|
| | ARAM Layers | Atrr-H | Atrr-D | C | F |
| Model I | 1 | 1 | 16 | 88.49% | 84.12% |
| Model II | 1 | 2 | 16 | 89.78% | 85.93% |
| Model III | 1 | 2 | 32 | 91.52% | 87.68% |
| Model IV | 2 | 4 | 32 | 94.38% | 90.79% |
| Model V | 2 | 4 | 64 | 95.59% | 92.91% |

We conclude that there are two possible reasons for this. On the one hand, the number of fine-grained categories is around eight times more than that of coarse-grained categories, which leads to more complex situation. On the other hand, the sample size of each fine-grained category is unbalanced, which causes significant differences in the accuracy of each category. Therefore, although we select categorical cross-entropy as the loss function, it is still necessary for us to balance the sample size of each fine-grained category during the training process. Nevertheless, such a data preprocessing approach is only theoretically feasible. This is because the question quantity gap between each category is extremely diverse in the existing datasets. Removing questions from certain categories to balance the data would result in inaccurate classification. POS-ARAN is also satisfactory from the point of view of generalization capability, as its accuracy in real scenarios is almost identical to its accuracy on the test set.

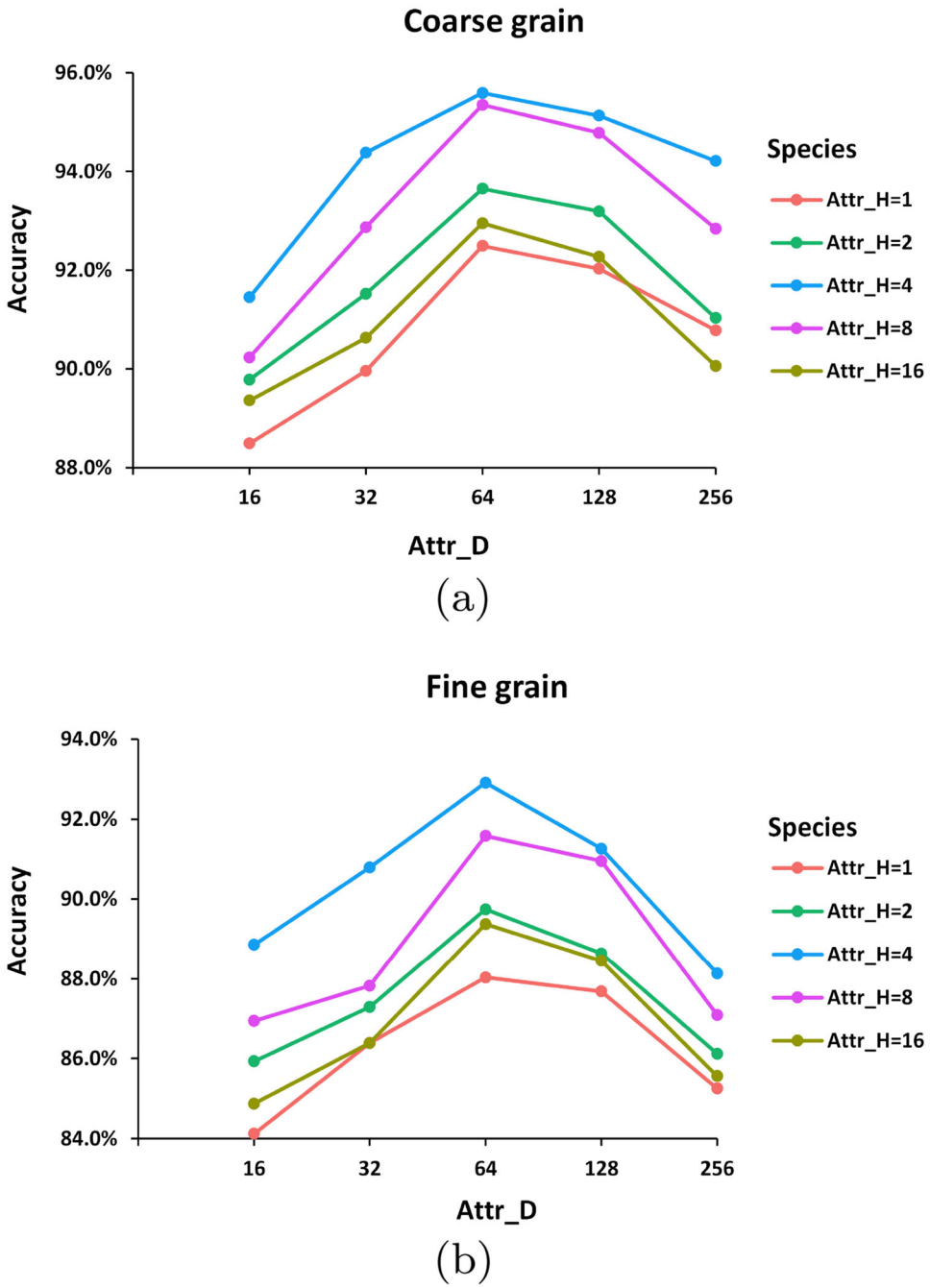


**Fig. 13** Results on EDQC with different Atrr-H and Atrr-D

## Conclusion

In this paper, we propose a POS-aware adjacent relation attention network (POS-ARAN) for question classification, which can capture both the long-term dependency and local representations in the text. To enhance the local representation of adjacent relations among words in different sentences, we introduce a learnable Gaussian bias which can obtain an adaptive self-attention distribution via a dynamic window to revise the original attention distribution. Furthermore, a novel POS-aware embedding layer is proposed, which helps to locate the appropriate headwords by syntactic information. The POS-ARAN applies an revised self-attention mechanism to classify questions and addresses the problem that the relation of neighboring words is weakened when calculating the weighted average of attention. The experiments on EDQC demonstrate that our POS-ARAN model exceeds most traditional and deep learning models in terms of performance and time. POS-ARAN achieves the coarse-grained accuracy of 95.59% and fine-grained accuracy of 92.91%, which proves that the POS-ARAN model is the more advanced model in question classification.

In the future, we plan to extend our research in the following aspects: (i) introducing SOTA language models into question classification to boost the overall performance with adequate contextual representation; (ii) integrating more syntactic cues into the question component extraction task to better capture semantic information; and (iii) investigating multi-task learning for question classification to apply to more complex situations.

**Funding** The funding has been received from National Key Research and Development Program of China with Grant no. 2021YFC2801001; Major Research plan of the National Social Science Foundation of China with Grant No.20&ZD130.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



## References

1. Yu X, Gong R, Chen P (2021) Question classification method in disease question answering system based on mcdplstm. In: 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 381–387. IEEE
2. Zulqarnain M, Alsaedi AKZ, Ghazali R, Ghouse MG, Sharif W, Husaini NA (2021) A comparative analysis on question classification task based on deep learning approaches. PeerJ Comput Sci 7:e570
3. Zhen L, Sun X (2021) The research of convolutional neural network based on integrated classification in question classification. Sci Program 2021
4. Mishra A, Patel D, Vijayakumar A, Li XL, Kapanipathi P, Talamadupula K (2021) Looking beyond sentence-level natural language inference for question answering and text summarization. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1322–1336
5. Soares MAC, Parreiras FS (2020) A literature review on question answering techniques, paradigms and systems. J King Saud Univ Comput Inform Sci 32(6):635–646
6. Tulbure AA, Tulbure AA, Dulf EH (2022) A review on modern defect detection models using dcnns-deep convolutional neural networks. J Adv Res 35:33–48
7. Zhang YD, Satapathy SC, Liu S, Li GR (2021) A five-layer deep convolutional neural network with stochastic pooling for chest ct-based covid-19 diagnosis. Mach Vis Appl 32(1):1–13
8. Fang W, Chen Y, Xue Q (2021) Survey on research of rnn-based spatio-temporal sequence prediction algorithms. J Big Data 3(3):97
9. Ren X, Gu H, Wei W (2021) Tree-rnn: tree structural recurrent neural network for network traffic classification. Expert Syst Appl 167:114,363
10. Gong P, Liu J, Yang Y, He H (2020) Towards knowledge enhanced language model for machine reading comprehension. IEEE Access 8:224,837-224,851
11. Bai Q, Zhou J, He L (2022) Pg-rnn: using position-gated recurrent neural networks for aspect-based sentiment classification. J Supercomput 78(3):4073–4094
12. Therasa M, Mathivanan G (2022) Arnn-qa: adaptive recurrent neural network with feature optimization for incremental learning-based question answering system. Appl Soft Comput:109029
13. Soni S, Chouhan SS, Rathore SS (2022) Textconvonet: a convolutional neural network based architecture for text classification
14. Tan C, Ren Y, Wang C (2023) An adaptive convolution with label embedding for text classification. Appl Intell 53(1):804–812
15. Liu J, Yang Y, Lv S, Wang J, Chen H (2019) Attention-based bigru-cnn for chinese question classification. J Ambient Intell Hum Comput:1–12
16. Ma Z, Li S, Zhang H, Li L, Liu J (2022) Hierarchical convolutional recurrent neural network for chinese text classification. In: Second International Conference on Sensors and Information Technology (ICSI 2022), vol. 12248, pp. 213–219. SPIE
17. Pan X, Ge C, Lu R, Song S, Chen G, Huang Z, Huang G (2022) On the integration of self-attention and convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 815–825
18. Shang S, Liu J, Yang Y (2020) Multi-layer transformer aggregation encoder for answer generation. IEEE Access 8:90410–90419
19. Liu Y, Li P, Hu X (2022) Combining context-relevant features with multi-stage attention network for short text classification. Comput Speech Lang 71
20. Zheng YF, Gao ZH, Shen J, Zhai XS (2022) Optimising automatic text classification approach in adaptive online collaborative discussion-a perspective of attention mechanism-based bi-lstm. IEEE Trans Learn Technol
21. Yang B, Tu Z, Wong DF, Meng F, Chao LS, Zhang T (2018) Modeling localness for self-attention networks. arXiv preprint arXiv:1810.10182
22. Liu J, Yang Y, He H (2020) Multi-level semantic representation enhancement network for relationship extraction. Neurocomputing 403:282–293
23. Thavareesan S, Mahesan S (2020) Word embedding-based part of speech tagging in tamil texts. In: 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), pp. 478–482. IEEE
24. Yadav A, Vishwakarma DK (2020) Sentiment analysis using deep learning architectures: a review. Artif Intell Rev 53(6):4335–4385
25. Yuan S, Zhang Y, Tang J, Hall W, Cabotà JB (2020) Expert finding in community question answering: a review. Artif Intell Rev 53(2):843–874
26. Cambazoglu BB, Sanderson M, Scholer F, Croft B (2021) A review of public datasets in question answering research. In: ACM SIGIR Forum, vol 54. ACM New York, NY, USA, pp 1–23
27. Raboanary T, Wang S, Keet CM (2022) Generating answerable questions from ontologies for educational exercises. In: Research Conference on Metadata and Semantics Research, pp. 28–40. Springer
28. Radev D, Fan W, Qi H, Wu H, Grewal A (2002) Probabilistic question answering on the web. In: Proceedings of the 11th International Conference on World Wide Web, p. 408–419. Association for Com-

puting Machinery, New York, NY, USA. https://doi.org/10.1145/511446.511500
29. Kwok C, Etzioni O, Weld DS (2001) Scaling question answering to the web. ACM Trans Inf Syst 19(3):242–262. https://doi.org/10.1145/502115.502117
30. Silva J, Coheur L, Mendes A, Wichert A (2011) From symbolic to sub-symbolic information in question classification. Artif Intell Rev 35:137–154. https://doi.org/10.1007/s10462-010-9188-4
31. Huang Z, Thint M, Qin Z (2008) Question classification using head words and their hypernyms. In: Proceedings of the 2008 Conference on empirical methods in natural language processing, pp. 927–936
32. Zhang D, Lee WS (2003) Question classification using support vector machines. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 26–32
33. Kariminejad N, Hosseinalizadeh M, Pourghasemi HR, Bernatek-Jakiel A, Campetella G, Ownegh M (2019) Evaluation of factors affecting gully headcut location using summary statistics and the maximum entropy model: Golestan province, ne Iran. Sci Total Environ 677:281–298
34. Le Nguyen M, Tri NT, Shimazu A (2007) Subtree mining for question classification problem. In: IJCAI, pp. 1695–1700
35. Li X, Roth D (2006) Learning question classifiers: the role of semantic information. Natl Lang Eng 12(3):229–250
36. Yilmaz S, Toklu S (2020) A deep learning analysis on question classification task using word2vec representations. Neural Comput Appl:1–20
37. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1181. https://www.aclweb.org/anthology/D14-1181
38. Zhou P, Qi Z, Zheng S, Xu J, Bao H, Xu B (2016) Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. arXiv preprint arXiv:1611.06639
39. Wu H, Cheng S, Wang Z, Zhang S, Yuan F (2020) Multi-task learning based on question–answering style reviews for aspect category classification and aspect term extraction on gpu clusters. Cluster Computing pp. 1 – 14
40. Cai R, Zhu B, Ji L, Hao T, Yan J, Liu W (2017) An cnn-lstm attention approach to understanding user query intent from online health communities. In: 2017 ieee international conference on data mining workshops (icdmw), pp. 430–437. IEEE
41. Liang Y, Li H, Guo B, Yu Z, Zheng X, Samtani S, Zeng DD (2021) Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification. Inform Sci 548:295–312
42. Im J, Cho S (2017) Distance-based self-attention network for natural language inference. arXiv preprint arXiv:1712.02047
43. Stigler SM (1982) A modest proposal: a new standard for the normal. Am Stat 36(2):137–138. https://doi.org/10.1080/00031305.1982.10482810. https://www.tandfonline.com/doi/abs/10.1080/00031305.1982.10482810
44. Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. Association for Computational Linguistics, Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1166. https://www.aclweb.org/anthology/D15-1166
45. Xiong R, Yang Y, He D, Zheng K, Zheng S, Xing C, Zhang H, Lan Y, Wang L, Liu T (2020) On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10,524–10,533. PMLR
46. Liu J, Lin L, Ren H, Gu M, Wang J, Youn G, Kim JU (2018) Building neural network language model with pos-based negative sampling and stochastic conjugate gradient descent. Soft Comput 22(20):6705–6717
47. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543
48. Zhang P, Cai Y, Chen J, Chen W, Song H (2019) Combining part-of-speech tags and self-attention mechanism for simile recognition. IEEE Access 7:163,864-163,876. https://doi.org/10.1109/ACCESS.2019.2951717
49. Zhu W, Yao T, Zhang W, Wei B (2019) Part-of-speech-based long short-term memory network for learning sentence representations. IEEE Access 7:51810–51816. https://doi.org/10.1109/ACCESS.2019.2911983
50. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser LU, Polosukhin I (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
51. Li X, Roth D (2002) Learning question classifiers. In: Proceedings of the 19th International Conference on Computational Linguistics - Volume 1, COLING '02, p. 1–7. Association for Computational Linguistics, USA. https://doi.org/10.3115/1072228.1072378
52. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
53. Dash P, Kisku DR, Gupta P, Sing JK (2022) Fast face detection using a unified architecture for unconstrained and infrared face images. Cogn Syst Res 74:18–38
54. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. IJCAI'16, p. 2873–2879. AAAI Press
55. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1179. https://www.aclweb.org/anthology/D14-1179
56. Chiu JP, Nichols E (2016) Named entity recognition with bidirectional lstm-cnns. Trans Assoc Comput Linguist 4:357–370
57. Devlin J, Chang MW, Lee K, Toutanova K (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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423
58. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942
59. Ma Y, Wang J, Ren Y, Zhang S, Li R (2021) A multi-granularity fusion neural network model for medical question classification. In: 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), pp. 487–492. IEEE