DUT-NLP at MEDIQA 2019:An Adversarial Multi-Task Network to Jointly Model Recognizing Question Entailment and Question Answering

Huiwei Zhou, Xuefei Li, Weihong Yao, Chengkun Lang, Shixian Ning
School of Computer Science and Technology
Dalian University of Technology
116024 Dalian, China
{zhouhuiwei, weihongy}@dlut.edu.cn
{lixuefei, kunkun, ningshixian}@mail.dlut.edu.cn

Abstract

In this paper, we propose a novel model called Adversarial Multi-Task Network (AMTN) for jointly modeling Recognizing Question Entailment (RQE) and medical Question Answering (QA) tasks. AMTN utilizes a pre-trained BioBERT model and an Interactive Transformer to learn the shared semantic representations across different task through parameter sharing mechanism. Meanwhile, an adversarial training strategy is introduced to separate the private features of each task from the shared representations. Experiments on BioNLP 2019 RQE and QA shared task datasets show that our model benefits from the shared representations of both tasks provided by multi-task learning and adversarial training, and obtains significant improvements upon the single-task models.

1 Introduction

With the rapid development of Internet and medical care, online health queries are increasing at a high rate. In 2012, 59% of U.S. adults looked for health information online[1]. However, it is always difficult for search engines to return relevant and trustworthy health information every time if the symptoms are not accurately described (Pletneva et al., 2012; Scantlebury et al., 2017). Therefore, many websites provide online doctor consultation services, which can answer questions or give advice from doctors or experts to the customers. Unfortunately, manually answering some simple queries or answering similar questions multiple times is quite time-consuming and wasteful. A Question Answering (QA) system that can automatically understand and answer the health care questions asked by customers is urgently needed (Wren, 2012).

To this end, BioNLP 2019 (Abacha et al., 2019) provides a series of challenging shared tasks, including: (1) Natural Language Inference (NLI) in the clinical domain; (2) Recognizing Question Entailment (RQE); (3) medical Question Answering (QA). This paper mainly focuses on RQE and QA task.

RQE task aims at identifying entailment relation between two questions in the context of QA (Abacha and Fushman, 2016), which can be represented as “a question Q1 entails a question Q2 if every answer to Q2 is also a complete or partial answer to Q1”.

QA task aims at automatically filtering and improving the ranking of automatically retrieved answers (Abacha and Fushman, 2019). There are two targets for QA: (1) determining whether the given sentence could answer the given question; (2) ranking all the right answers according to their relevance to the question.

Neural networks and deep learning (DL) currently provide the best solutions for RQE and QA tasks. Among various neural networks, such as traditional Convolutional Neural Networks (CNN) (LeCun et al., 1998) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017) has demonstrated superiority in multiple

[1] http://www.pewinternet.org/2013/01/15/health-online-2013/

Proceedings of the BioNLP 2019 workshop, pages 437–445
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

natural language processing tasks (Verga et al., 2017; Shen et al., 2017; Yu et al., 2018). Transformer (Vaswani et al., 2017) is based solely on attention mechanisms and it can effectively capture the long-range dependencies between words.

More recently, the pre-trained language models, such as ELMo (Matthew et al., 2018), OpenAI GPT[2], and BERT (Devlin et al., 2018), have shown their effectiveness to capture the deep semantic and syntactic information of words. BioBERT (Lee et al., 2019) is one of the BERT-based pre-trained language model for biomedical domain, and it achieves great improvement in many biomedical tasks. For this reason, we believe that the pre-trained language models, especially the BioBERT, should be valid for RQE and QA under reasonable use.

Most previous researches train the model of RQE task or QA task separately based on a single training set. However, such single-task method cannot provide essential mutual supports between the two tasks. The inherent interactions between the two tasks might help us do even better on the RQE and QA tasks. RQE task can find Frequently Asked Questions (FAQs) similar to a consumer health question, providing consumers with appropriate FAQs and enabling QA systems to identify the right answers with greater precision and higher speed (Harabagiu and Hickl, 2006).

Multi-Task Learning (MTL) is a learning paradigm in machine learning and its aim is to leverage shared representations contained in multiple related tasks to help improve the generalization performance of all the tasks. MTL is usually done with parameter sharing of hidden layers. Hard parameter sharing is the most commonly used approach to MTL in neural networks. It is generally applied by sharing the hidden layers between all tasks, while keeping several task-specific output layers. However, it is difficult for MTL to distinguish the commonalities and differences between different tasks.

A common way to improve the robustness of the system is to train the system using different datasets through adversarial training (Goodfellow et al., 2014). Chen et al. (2017) propose a shared-private model, which extracts shared features and private features from multiple corpus, and introduces adversarial training for shared representation learning. Drawing on the practices of previous studies, we plan to use an adversarial multi-task framework to extract the noise-robust shared representation directly.

Considering the similarity between RQE and QA tasks, this paper proposes a novel Adversarial Multi-Task Network (AMTN) to jointly model these two tasks. Specifically, AMTN first utilizes BioBERT as an embedding layer to generate context-dependent word representations. Then, a common Interactive Transformer layer is introduced for sentence representation learning and inter-sentence relationship modeling, which allows knowledge transfer from other tasks. Finally, two specific classifiers are used for RQE and QA tasks respectively. Here, we only consider the target (1) of QA task for the multi-task learning to ensure the consistency between RQE and QA tasks. Furthermore, to prevent the shared and private feature spaces from interfering with each other, an adversarial training strategy is introduced to make the shared feature representations to be more compatible and task-invariant among different tasks. Experimental results show that our AMTN model is effective to improve the performance for both RQE and QA tasks upon the single-task models, which demonstrates the superiority of the adversarial multi-task strategy.

Our contributions can be summarized into two folds.

- A well-designed Interactive Transformer layer is introduced for sentence representation learning and inter-sentence relationship modeling.

- A novel adversarial multi-task strategy is introduced to jointly model RQE and QA tasks, in which multi-task learning is proposed for shared representation learning and adversarial training is used to force the shared representation purer and task-invariant.

2 Method

This section gives a detailed description of the proposed AMTN, which is shown in Figure 1. AMTN mainly consists of three parts: a shared

[2] https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

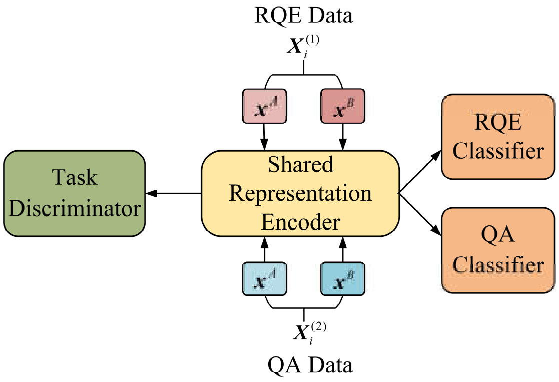

Figure 1: The framework of AMTN.

encoder, a task discriminator and two classifiers for the RQE task and the QA task, respectively.

Shared encoder is used to learn the shared semantic representations across different tasks through parameter sharing mechanism. Task discriminator is used to form an adversarial training with the shared encoder to separate the private features of each task from the shared representations. Two specific classifiers are applied to judge whether a sentence pair is an entailment relationship (RQE task) or a question-and-answer relationship (QA task).

Next, we will use four subsections to introduce our AMTN model in detail: Data Preprocessing, Shared Representation Learning, Task Specific Classifier and Adversarial Training.

2.1 Data Preprocessing

Define a data set $\{\boldsymbol{X}_i^{(k)}, \mathbf{y}_i^{(k)}\}_{i=1}^{N_k}$, where $\boldsymbol{X}_i^{(k)}$ is the i^{th} input for the k^{th} task, $\mathbf{y}_i^{(k)}$ is the corresponding labels of $\boldsymbol{X}_i^{(k)}$, N_k is the number of training data in the k^{th} task. In this paper, $k=1$ refers to RQE task, and $k=2$ refers to QA task. Each $\boldsymbol{X}_i^{(k)}$ is composed of the concatenation of a unique [CLS] flag with a sentence pair $\boldsymbol{x}^A = \left\{x_1^A, x_2^A, \ldots, x_n^A\right\}$ and $\boldsymbol{x}^B = \left\{x_1^B, x_2^B, \ldots, x_m^B\right\}$, where n, m are the sequence lengths. Specially, since the answers of the QA task are too long, we intercept the first sentence of them as $\boldsymbol{x}^B$.

2.2 Shared Representation Learning

We use the shared encoder to learn the shared representations as the input for the classifiers and the task discriminator. Figure 2 illustrates the architecture of shared encoder, which contains BioBERT Embedding Layer, Interactive Transformer Layer and Combination Layer.

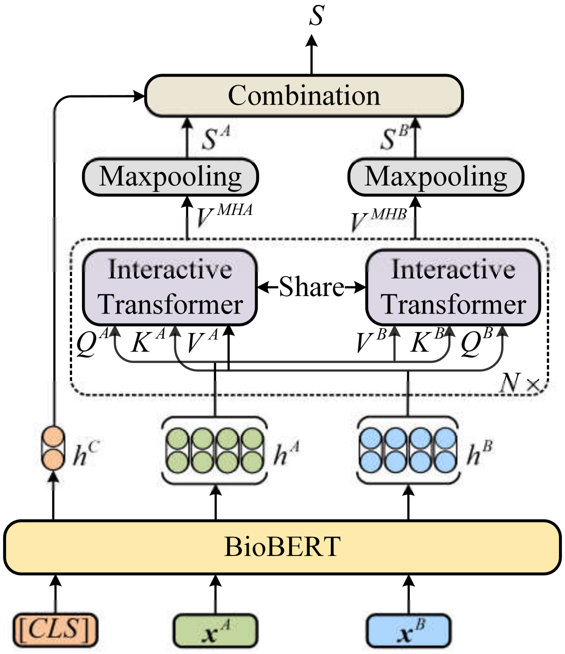

Figure 2: The architecture of the shared encoder.

BioBERT Embedding Layer: BioBERT is a domain specific language representation model pre-trained on large-scale biomedical corpora (Lee et al., 2019). It could effectively enhance the learning ability of encoding biomedical information.

We use BioBERT as an embedding layer, whose final hidden representation of each word is treated as word embedding. Given the sequence input $\boldsymbol{X}$, the corresponding hidden representation sequence $\boldsymbol{H} = \{\boldsymbol{h}^C, \boldsymbol{h}^A, \boldsymbol{h}^B\}$ can be obtained through the BioBERT layer, where $\boldsymbol{h}^A \in \mathbb{R}^{d_1 \times n}$, $\boldsymbol{h}^B \in \mathbb{R}^{d_1 \times m}$ and $\boldsymbol{h}^C \in \mathbb{R}^{d_1 \times 1}$ correspond to the sentence $\boldsymbol{x}^A$, the sentence $\boldsymbol{x}^B$ and the unique [CLS] flag respectively, and d_1 is the output dimension of BioBERT.

Interactive Transformer Layer: To effectively capture the long dependency information and establish an interaction between the two sentences, the hidden representation sequences $\boldsymbol{h}^A$ and $\boldsymbol{h}^B$ are fed to an Interactive Transformer Layer. The Interactive Transformer consists of N blocks, each of which contains a multi-head attention with interactive process. Multi-head attention performs the scaled dot product attention multiple times on linearly projected query (Q), Key (K) and Value (V), which is shown in the following formula:

$$\text{Attention}(Q,K,V)=\text{softmax}(\frac{Q^TK}{\sqrt{d_K}})V \qquad (1)$$

where d_K is the dimension of K. Vaswani et al. (2017) point out that the input of softmax grows large in magnitude, pushing the softmax function into regions where it has extremely small gradients. Therefore, the dot productions are scaled by $1/\sqrt{d_K}$ to counteract this effect.

For the first sentence $\boldsymbol{x}^A$, we take its hidden representation $\boldsymbol{h}^A$ as Q and $\boldsymbol{h}^B$ as K, V. In this way, the information flow inside features of the sentence $\boldsymbol{x}^B$ are dynamically conditioned on the features of the sentence $\boldsymbol{x}^A$. The inputs for the first sentence $\boldsymbol{x}^A$ can be represented as:

$$\begin{aligned} Q^A &= \left\{h_1^A, h_2^A, ..., h_n^A\right\}, \\ K^A &= \left\{h_1^B, h_2^B, ..., h_m^B\right\}, \\ V^A &= \left\{h_1^B, h_2^B, ..., h_m^B\right\}. \end{aligned} \qquad (2)$$

For the second sentence $\boldsymbol{x}^B$, we take $\boldsymbol{h}^B$ as Q and take $\boldsymbol{h}^A$ as K, V, which can be represented as:

$$\begin{aligned} Q^B &= \left\{h_1^B, h_2^B, ..., h_m^B\right\}, \\ K^B &= \left\{h_1^A, h_2^A, ..., h_n^A\right\}, \\ V^B &= \left\{h_1^A, h_2^A, ..., h_n^A\right\}。 \end{aligned} \qquad (3)$$

Therefore, the multi-head attentions with interactive process for the given pair of sentences can be formulated as:

$$head_l^A = Attention(\mathbf{W}_l^Q Q^A, \mathbf{W}_l^K K^A, \mathbf{W}_l^V V^A) \qquad (4)$$

$$head_l^B = Attention(\mathbf{W}_l^Q Q^B, \mathbf{W}_l^K K^B, \mathbf{W}_l^V V^B) \qquad (5)$$

$$V^{MHA} = \mathbf{W}^H[head_1^A; head_2^A; ...; head_L^A] \qquad (6)$$

$$V^{MHB} = \mathbf{W}^H[head_1^B; head_2^B; ...; head_L^B] \qquad (7)$$

where $\mathbf{W}_l^Q \in \mathbb{R}^{d_{head} \times d_1}$, $\mathbf{W}_l^K \in \mathbb{R}^{d_{head} \times d_1}$, $\mathbf{W}_l^V \in \mathbb{R}^{d_{head} \times d_1}$ and $\mathbf{W}^H \in \mathbb{R}^{d_1 \times Ld_{head}}$ are trainable shared parameters. $[head_1; head_2; ...; head_L]$ is a concatenation of outputs of L heads.

Different from the original Transformer, in which the input of Q, K and V are all the same, Interactive Transformer takes different sentences as the inputs of Q and K, V. In this way, we expect to effectively compute dependencies between any two words of the sentence pairs and encode the abundant semantic information for each sequence word.

Combination Layer: After modeling the association between the two sentences, we utilize the max pooling operation to obtain the final shared semantic representations of $\boldsymbol{x}^A$ and $\boldsymbol{x}^B$ respectively:

$$S^A = Maxpooling(V^{MHA}) \qquad (8)$$

$$S^B = Maxpooling(V^{MHB}) \qquad (9)$$

Then, we perform vector combination on S^A, S^B, and flag representation $\boldsymbol{h}^C$ through a dense layer to generate the sentence pair representation S for classification, which is calculated as follows:

$$\begin{aligned} S = \text{ReLU}(&\mathbf{W}_0[S^A; S^B; S^A - S^B; \\ &S^A \odot S^B; \boldsymbol{h}^C] + b_0) \end{aligned} \qquad (10)$$

where $\mathbf{W}_0 \in \mathbb{R}^{d_0 \times 5d_1}$ and $b_0 \in \mathbb{R}^{d_0 \times 1}$ are trainable parameters.

2.3 Task Specific Classifier

For each task, a specific classifier is employed to judge whether a sentence pair is an entailment relationship (RQE) or a question-and-answer relationship (QA). Each classifier is composed of a two-layer fully-connected neural network, which uses a ReLU nonlinearity after the first fully connected layer and a softmax nonlinearity after the second fully connected layer. It can be written as follows:

$$\hat{\mathbf{y}}_i^{(m)} = softmax\left(\mathbf{W}_2 \text{ReLU}\left(\mathbf{W}_1 S + b_1\right) + b_2\right) \qquad (11)$$

The classifier takes the sentence pair representation S as input and outputs a probability distribution to predict whether the current sentence pair is entailment relation or question-and-answer relation.

Both classifiers are trained by optimizing the cross-entropy loss as follows:

$$J_{classifier} = -\sum_{i=1}^{N_k} \sum_{j=1}^{C} \left(\mathbf{y}_{i,j}^{(k)} \log\left(\hat{\mathbf{y}}_{i,j}^{(k)}\right)\right) \qquad (12)$$

where C is the number of categories of classification label, $\hat{\mathbf{y}}_{i,j}$ is the predicted probability of the j^{th} category of the i^{th} sentence pair.

2.4 Adversarial Training

In order to make shared representations contain more common information and reduce the mixing of task-specific information, adversarial training is introduced into the above multi-task framework.

The goal of the proposed adversarial training strategy is to form an adversary with shared representation learning by introducing a task discriminator. In this paper, we take the shared encoder as generative network G and the task discriminator as discriminative model D , in which G needs to learn as much semantic information as possible from the shared data distribution between the two tasks and D aims to determine which task (RQE or QA) the input sentence belongs to by using the shared representations.

Specifically, we first use the shared encoder $\mathrm{G}(\boldsymbol{X},\theta_s)$, which is mentioned in section 2.2, to get the sentence pair representation S . θ_s is the shared parameter need to be trained. Then, the shared representations will be fed to the task discriminator D to determine the task to which the current input belongs. D can be expressed by the following formula:

$$\mathrm{D}\left(S,\theta_d\right) = \mathit{softmax}\left(\mathbf{W}_4\,\mathrm{ReLU}\left(\mathbf{W}_3 S + b_3\right) + b_4\right) \quad (13)$$

Besides the task loss for RQE and QA, we additionally introduce an adversarial loss J_{adv} to prevent task-specific feature from creeping into shared space and thus get a purer shared representation. The adversarial loss J_{adv} is trained in alternating fashion as shown below:

$$J_{adv} = \min_{\theta_s}\left(\max_{\theta_d}\left(\sum_{k=1}^{2}\sum_{i=1}^{N_k}\mathbf{t}_i^{(k)}\log\left[\mathrm{D}\left(\mathrm{G}\left(\boldsymbol{X},\theta_s\right),\theta_d\right)\right]\right)\right) \quad (14)$$

where $\mathbf{t}_i^{(k)}$ is the correct task label (RQE task or QA task) of the given sentence pair $\boldsymbol{X}$. Here the basic idea is that, the shared representations learned by the shared encoder need to mislead the task discriminator. At the same time, the task discriminator needs to predict the task (RQE or QA) to which the data belongs as accurately as possible. The two are adversarial to each other and alternately optimized to separate the private features from the shared representations.

Finally, the shared encoder and the task discriminator reach a balance point and achieve mutual promotion.

3 Experiments

3.1 Dataset

Our experiments are conducted on the BioNLP RQE and QA shared tasks. The QA dataset contains a total of 3042 question-answer pairs: 1701 for training, 234 for validation, and 1107 for test. The RQE dataset contains a total of 9120 question pairs: 8588 for training, 302 for validation, and 230 for test. The statistic of the two datasets are shown in Table 1.

Task	Train	Validation	Test
RQE	8588	302	230
QA	1701	234	1107

Table 1: Statistic of sentence pairs in RQE and QA datasets.

3.2 Experimental settings and metric

In the shared encoder module, we use the pre-trained uncased BioBERT$_{\mathrm{base}}$[3] for computational complexity considerations. The number of its Transformer blocks and multiple heads are both 12. For the Interactive Transformer, we use 3 blocks with 16 heads. The hidden layer dimension of BioBERT$_{\mathrm{base}}$ and Interactive Transformer are both set to 768. We use a mini batch size of 8 and epoch of 30. Adam optimizer (Kingma and Ba, 2014) is used for both shared encoder and task discriminator to tune the parameters at the learning rates of $\lambda_1 = 1\mathrm{e}-5$ and $\lambda_2 = 1\mathrm{e}-4$, respectively. Specially, due to the small quantity of QA training data, we oversample it three times during training in order to balance the dataset of two tasks. The hyper-parameters settings used in this paper are shown in Table 2. The performance RQE and QA tasks are evaluated by the official evaluation scripts[4], which adopt accuracy as the evaluation metric.

[3] https://github.com/naver/biobert-pretrained

[4] https://github.com/abachaa/MEDIQA2019/tree/master/Eval_Scripts

Hyper-parameters	Value
Pre-trained Model Heads	12
Pre-trained Model Blocks	12
Interactive Transformer Heads	16
Interactive Transformer Blocks	3
Hidden Layer Dimension	768
Epoch	30
Mini-batch	8
Learning Rate for Shared Encoder	1e-5
Learning Rate for Discriminator	1e-4

Table 2: Hyper-parameters settings.

3.3 Effects of the Adversarial Multi-Task Learning Strategy

This section first proposes two baseline strategies for comparison as described below:

- **Multi-Task**: Under this strategy, the architecture is constructed by removing the discriminator D from AMTN. We call it AMTN-Discriminator.

- **Single-Task**: Under this strategy, the architecture is constructed by removing the discriminator D from AMTN, and using the same classifier for the two tasks. We call it Single-Task Network (STN).

The results are shown in Table 3. From the table, we can see that single-task learning achieves the worst results, which is probably due to the simple model architecture. For the three methods using different dataset in Single-task learning, STN (QA+RQE) performs better than STN (QA) and STN (RQE). It demonstrates that the two datasets have quite similar information distributions that could adequately complement each other and contribute to both RQE and QA tasks.

From the second block in Table 4, we can see that **Multi-Task** strategy performs clearly better than **Single-Task**. Note that, AMTN-Discriminator has an accuracy rate of 63.6% and 74.5% for RQE and QA tasks, which is the result of our submission in the task website. Multi-task learning jointly trains multiple sub-task models through a shared encoder. It can effectively capture the common features of the two task data, thereby promoting the generalization ability of RQE and QA tasks synchronously.

To explore the effects of the proposed adversarial multi-task strategy. Furthermore, we arm the above **Multi-Task** strategy with adversarial training, i.e. adding a discriminator to form the adversary with shared representation learning:

- **Adversarial Multi-Task**: Under this strategy, two architectures are constructed. One is our proposed AMTN. The other is a variant of AMTN, which adds a Private Encoder for each task to parallelly learn task-specific representations and shared representations.

Table 4 lists the comparison results. Compared with **Multi-Task**. AMTN achieves further improvement (0.7% and 1.3% accuracy for RQE and QA tasks respectively) with the help of additional task discriminator and the introduction

Strategy	Architecture	RQE	QA
Single-Task	STN (QA)	59.1	71.4
	STN (RQE)	50.0	61.0
	STN (QA+RQE)	61.7	71.4
Multi-Task	AMTN-Discriminator[†]	63.6	74.5

Table 3: Effects of multi-task learning strategy. All the results are reported by accuracy (%). STN (QA), STN (RQE) and STN (QA+RQE) represent STN trained on QA dataset, QRE dataset and both datasets, respectively. [†] indicates our submission results.

Strategy	Architecture	RQE	QA
Multi-Task	AMTN-Discriminator[†]	63.6	74.5
Adversarial Multi-Task	AMTN	**64.3**	**75.8**
	AMTN+Private Encoder	58.3	72.5

Table 4: Effects of adversarial multi-task learning strategy. All the results are reported by accuracy (%). [†] indicates our submission results. Bold font indicates the best performance.

of adversarial loss. We believe that the discriminator could strip private features from shared representations and make shared representations more general.

Finally, when we add a private encoder for each task, i.e. AMTN+Private Encoder, we can see that the performance is significantly reduced by 6.0% and 3.3% accuracy in RQE and QA tasks, respectively. Although private representation could provide task-specific information, it will introduce too many redundant parameters that could make the model prone to over-fitting, resulting in performance degradation.

3.4 Effects of the Shared Encoder

Our AMTN model uses **Interactive Transformer** as shared encoder to perform shared representation learning. To verify the effects of the shared encoder, we compare the **Interactive Transformer** with the following three baseline methods:

- **CNN** encoder: This method uses Convolutional Neural Network (CNN) to encode each sentence. 256 filters with window size of 3,4,5 are used in CNN, respectively.
- **Bi-LSTM** encoder: This method uses a single-layer bidirectional Long Short-Term Memory network (Bi-LSTM) to encode each sentence. The hidden layer dimension of each direction is set to 384.
- **Transformer** encoder: This method uses an original Transformer to encode each sentence. For sentence $\boldsymbol{x}^A$, the three input (Q, K and V) of Transformer are all $\boldsymbol{h}^A$. For sentence $\boldsymbol{x}^B$, the three input (Q, K and V) of Transformer are all $\boldsymbol{h}^B$. That is to say, there is no interaction between the two sentences in this encoder.

Note that, the final sentence representation is generated by max pooling on the output of the above shared encoder.

In addition, previous works in biomedical RQE and QA often use word embeddings trained on PubMed or PMC corpus. To verify the superiority of pre-trained language representation model, the above four shared encoders (including **Interactive Transformer**) are respectively equipped with the following three word representation methods:

- Word2Vec: Each word in a sentence is represented by word embeddings trained on PubMed abstracts and PubMed Central full-text articles (Wei et al., 2013) with Word2vec toolkit (Mikolov et al., 2013). The dimension of the pre-trained word embedding is 100. We use a transition matrix to convert its dimension to 768.
- BERT: The pre-trained BERT model is used to generate a hidden representation h of each word in the sentence as its word embedding. The purpose of this method is to increase the generalization ability of the Word2Vec and fully describe the character level, word level and sentence level information and even the relationship between sentences.
- BioBERT: Same as above, the pre-trained BioBERT model is used to generate a hidden representation h of each word in the sentence as its word embedding.

Encoder	Embedding	RQE	QA
CNN	Word2vec	57.4	56.5
	BERT	60.8	73.7
	BioBERT	63.0	74.9
Bi-LSTM	Word2vec	52.6	58.4
	BERT	62.2	74.1
	BioBERT	62.2	75.1
Transformer	Word2vec	54.4	60.8
	BERT	59.6	72.7
	BioBERT	59.6	74.0
Interactive Transformer	Word2vec	57.8	62.3
	BERT	61.7	73.5
	BioBERT	**64.3**	**75.8**

Table 5: Effects of shared encoder. All the results are reported by accuracy (%). Bold font indicates the best performance.

Table 5 lists all the results on the RQE and QA dataset. By analyzing Table 5, we obtain the following conclusions. On the one hand, we can find that BERT brings a qualitative leap to the performance of both RQE and QA tasks upon Wor2Vec. BioBERT enriches BERT with a large amount of biomedical information and achieves approximately 1% absolute accuracy improvement over the BERT on both the tasks. It

Error Type	Sentence Pair	Task	Gold/Prediction
Acronyms	Q1: … he went to hospital to have medical check-up with endoscopic ultrasonography, and found **GIST** with about 1cm in size … What are we supposed to do? … Q2: What are the treatments for **Gastrointestinal Stromal Tumor**?	RQE	Entailment/Contradiction
Ambiguous Samples	Q: Spina bifida; vertbral fusion; syrinx tethered cord. Can u help for treatment of these problem? A: Spina bifida (Complications): Spina bifida may cause minimal symptoms or only minor physical disabilities.	QA	True/False
Semantic Confusion	Q1: **What is** the **possibility** of **atypical pneumonia** occurring again less than a month after **treatment**? Q2: **What are** the **possible treatments** for **atypical pneumonia**?	RQE	Contradiction/Entailment

Table 6: Failure cases predicted by AMTN.

shows that pre-trained models could improve model robustness and uncertainty estimates.

On the other hand, among the four different encoders, **Interactive Transformer** shows the best results overall. **Interactive Transformer** could not only capture the long-range dependency information, but also establish an interaction between the given two sentences by the interactive process. The benefit of introducing the interactive process is that it can efficiently compute dependencies between any two words in a sentence pair and encode the rich semantic information for each sequence word.

3.5 Error Analysis

Although the proposed AMTN achieves great performance over strong baselines, some failure cases are also observed. We have carried out detailed statistics and analysis of these errors, and classified the possible causes into the following three categories.

The first error type is acronyms. Since most biomedical concepts have acronyms, e.g. "Gastrointestinal Stromal Tumor" vs. "GIST" in first sentence pair in Table 6, it is quite difficult for model to determine whether the two sentences focus on the same topic without any external knowledge, thus resulting in misclassification. This problem is also our concern for future work, e.g. how to integrate prior knowledge into the model.

The second error type is ambiguous samples, which means that the relationship between the sentences is fuzzy and difficult to judge, such as the QA sentence pair shown in the second block of Table 5. Its golden label is True, however, the answer sentence seems to be irrelevant to the question, thus leads to the wrong classification of our model.

The third error type is semantic confusion, which refers to the semantic misunderstanding caused by complex syntax or collocation of phrases. Take the sentence pair in third block of Table 6 as an example: Q1 contains almost all the words in Q2 ("possible", "atypical pneumonia", "treatments" and etc.), while the two sentences are of Contradiction relation. We believe that the sentence pair is quite confusing that AMTN does not really "understand" it.

4 Conclusion

In this paper, we propose an Adversarial Multi-Task Network to jointly model RQE and QA shared tasks. ATMN employs BioBERT and Interactive Transformer as the shared encoder to learn the shared representations across the two tasks. A discriminator is further introduced to form an adversarial training with the shared encoder for purer shared semantic representations. Experiments on BioNLP 2019 RQE and QA shared tasks show that our proposed AMTN model benefits from the shared representations of both tasks provided by multi-task learning and adversarial training, and gains a significant improvement upon the single-task models.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61772109) and the Ministry of education of Humanities and Social Science research and planning Fund of China (No. 17YJA740076).

References

Natalia Pletneva, Alejandro Vargas, Konstantina Kalogianni, and Celia Boyer. 2012. Online health information search: what struggles and empowers the users? Results of an online survey. *Studies in health technology and informatics*, 180:843-847.

Arabella Scantlebury, Alison M. Booth, and Bec Hanley. 2017. Experiences, practices and barriers to accessing health information: a qualitative study. *International Journal of Medical Informatics*, 103:103-8.

Jonathan D. Wren. 2011. Question answering systems in biology and medicine--the time is now. *Bioinformatics*, 27(14):2025-2026.

Asma B. Abacha, Chaitanya Shivade, and Dina D. Fushman. 2019. Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. *In Proceedings of the MEDIQA 2019.*Asma B. Abacha and Dina D. Fushman. 2016. Recognizing question entailment for medical question answering. AMIA. In *Proceedings of the AMIA Symposium*, pages 310-318.

Asma B. Abacha and Dina D. 2019. A question-entailment approach to question answering. *arXiv preprint arXiv:1901.08079*.

Yann LeCun, Leon Bottou, Yoshua Bengio, and PatrickHaffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278-2324.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735-1780.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Patrick Verga, Emma Strubell, Ofer Shai, and Andrew McCallum. 2017. Attending to all mention pairs for full abstract biological relation extraction. *arXiv preprint arXiv:1710.08312*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. DiSAN: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.048055*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: A pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 905-912.

Rich Caruana. 1998. Multitask Learning. *Autonomous Agents and Multi-Agent Systems*, 27(1):95–133.

Ian Goodfellow, Jean P. Abadie, Mehdi Mirza, Bing Xu, David W. Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-Criteria Learning for Chinese Word Segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193-1203.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. PubTator: A web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41(W1):W518-W522.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.