Technical Question Answering across Tasks and Domains

Wenhao Yu[†], Lingfei Wu[‡], Yu Deng[‡], Qingkai Zeng[†],
Ruchi Mahindru[‡], Sinem Guven[‡], Meng Jiang[†]
†University of Notre Dame, Notre Dame, IN, USA
‡IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
†{wyu1, qzeng, mjiang2}@nd.edu
‡{wuli, dengy, rmahindr, sguven}@us.ibm.com

Abstract

Building automatic technical support system is an important yet challenge task. Conceptually, to answer a user question on a technical forum, a human expert has to first retrieve relevant documents, and then read them carefully to identify the answer snippet. Despite huge success the researchers have achieved in coping with general domain question answering (QA), much less attentions have been paid for investigating technical QA. Specifically, existing methods suffer from several unique challenges (i) the question and answer rarely overlaps substantially and (ii) very limited data size. In this paper, we propose a novel framework of deep transfer learning to effectively address technical QA across tasks and domains. To this end, we present an adjustable joint learning approach for document retrieval and reading comprehension tasks. Our experiments on the TechQA demonstrates superior performance compared with state-of-the-art methods.

1 Introduction

Recent years have seen a surge of interests in building automatic technical support system, partially due to high cost of training and maintaining human experts and significant difficulty in providing timely responses during the peak season. Huge successes have been achieved in coping with open-domain QA tasks (Chen and Yih, 2020), especially with advancement of large pre-training language models (Devlin et al., 2019). Among them, two-stage retrieve-then-read framework is the mainstream way to solve open-domain QA tasks, pioneered by (Chen et al., 2017): a retriever component finding a document that might contain an answer from a large collection of documents, followed by a reader component finding the answer snippet in a given paragraph or a document. Recently, various pre-training language models (e.g., BERT) have dominated the encoder design for solving different open-domain QA tasks (Karpukhin

(a) A factoid QA example in the SQuAD dataset.

(b) A non-factoid QA example in the TechQA dataset.

Figure 1: Factoid QA is semantic aligned but non-factoid QA has few overlapping words. Semantic similarities between such non-factoid QA is not indicative.

et al., 2020; Xiong et al., 2020).

Despite the tremendous successes achieved in general QA domain, technical QA have not yet been well investigated due to several unique challenges. First, technical QAs are non-factoid. The question and answer can hardly overlap substantially, because the answer typically fills in missing information and actionable solutions to the question such as steps for installing a software package and configuring an application. Different from factoid questions that are typically aligned with a span of text in document (Rajpurkar et al., 2016, 2018), semantic similarities between such non-factoid QA pairs could have a large gap as shown in Fig.1. Therefore, the retrieval module in retrieve-then-read framework might find documents that do not contain correct answers due to the semantic gap in non-factoid QAs (Karpukhin et al., 2020; Lee et al., 2019; Yu et al., 2020b). Second, compared to SQuAD (with more than 100,000 QA pairs), technical domain datasets typically have a much smaller number of labelled QA pairs (e.g., about 1,400 in TechQA), partially due to the prohibitive cost of creating labelled data. In addition, there are limited real user questions and technical support documents, especially for some new tech products and

178

communities. Since the pre-trained language models are mainly trained on general domain corpora, directly fine-tuning pre-trained language models may lead to unsatisfying performance due to the large discrepancy between source tasks (general domains) and target tasks (technical domains) (Chang et al., 2020; Gururangan et al., 2020).

To address the aforementioned challenges, we propose a novel deep transfer learning framework that explores knowledge transfer across tasks and domains (TransTD). TransTD consists of two components: TransT (knowledge transfer across tasks) and TransD (knowledge transfer across domains). TransTD jointly learns snippet prediction (reading comprehension) task and matching prediction (document retrieval) task simultaneously, applying it on both general domain QA and target domain QA.

To address the first challenge of non-factoid QAs, TransT leverages a joint learning model that directly ranks all predicted snippets by reading each pair of query and candidate document. It optimizes matching prediction and snippet prediction in parallel. Compared to two-stage retrieve-then-read methods that only read most semantically related documents, TransT considers potential snippets in every candidate document. When jointly training these two tasks, snippet prediction pays attention to local correspondence and matching prediction helps understand the semantic relationship from a global perspective, allowing the multi-head attentions in BERT-based encoders to jointly attend to information from different representation subspaces at different positions. Besides, the weights of two training objectives can be dynamically learned to pay more attention on the more difficult task when training different data samples.

To address the second challenge of learning with limited data, TransD leverages a deep transfer learning model to transfer knowledge from general domain QAs to technical domain QAs. General domain QA dataset like SQuAD has a much larger data size and a similar task setting (i.e., snippet prediction). Though knowledge is different between two domains, by learning the ability to answer questions in general domains, the model can quickly adapt and learn efficiently when changing into a new domain, reflected in faster convergence and better performance. Transfer learning helps avoid overfitting on technical QAs with limited size of data. Specifically, our model first applies the multi-task joint learning in general domain QAs (SQuAD), then transfers model parameters to initialize the training in the target domain QAs (TechQA), making knowledge transfer across domains to address data limitation.

We conducted extensive experiments on the TechQA dataset and utilized BERT as basic models. Experiments show that TransTD can provide superior performance than models with no knowledge transfer and other state-of-the-art methods.

2 Related Work

Open-Domain QA Open-domain textual question answering is a task that requires a system to answer factoid questions using a large collection of documents as the information source, without the need of pre-specifying topics or domains (Chen and Yih, 2020). Two-stage retriever-reader framework is the mainstream way to solve open-domain QA, pioneered by (Chen et al., 2017). Recent work has improved this two-stage open-domain QA from different perspectives such as novel pre-training methods (Lee et al., 2019; Guu et al., 2020), semantic alignment between question and passage (Lee et al., 2019; Karpukhin et al., 2020; Wu et al., 2018), cross-attention based BERT retriever (Yang et al., 2019; Gardner et al., 2019), global normalization between multiple passages (Wang et al., 2019).

Transfer Learning Transfer learning studies how to transfer knowledge from auxiliary domains to a target domain (Pan and Yang, 2009; Jiang et al., 2015; Yao et al., 2019). Recent advances of deep learning technologies with transfer learning has achieved great success in a variety of NLP tasks (Ruder et al., 2019). Several research work in this domain greatly enrich the application and technology of transfer learning on question answering from different perspectives (Min et al., 2017; Deng et al., 2018; Castelli et al., 2020; Yu et al., 2020a). Although transfer learning has been successfully applied to various QA applications, its applicability to technical QA has yet to be investigated. In this work, we focus on leveraging transfer learning to enhance QA in tech domain.

3 Research Problem

In the technical support domain, suppose we have a set of questions $\mathcal{Q}$ and a large collection of documents $\mathcal{D}$. For each question $Q \in \mathcal{Q}$, we aim at finding a relevant document $D \in \mathcal{D}$ and extracting the snippet answer $S = (D_{start}, D_{end})$ in the

document D. Note that the answer may not exist, and so, the relevant document may not exist, either. All predicted snippets are ranked by a specific span score calculation method, and (usually) the top-1[1] answer span is chosen to answer the given question.

4 Proposed Framework

In this section, we present our proposed framework for technical QA. Given a query, we first obtain 50 Technotes by issuing the query to the search engine Elasticsearch[2]. Instead of using a document retriever based on semantic similarity between the query and each document, our proposed TransTD jointly optimizes snippet prediction and matching prediction in a parallel style. Figure 2 illustrates the design of the framework. It has a multi-task learning method to transfer knowledge across the snippet prediction (reading comprehension) and matching prediction (document retrieval) tasks. This method is further applied to pre-train the model on auxiliary domain QAs[3]. Furthermore, the weights of two training objectives are dynamically adjusted by calculating the difference between real answer snippet and predicted snippet. So, the model can focus on optimizing the more difficult task when training different data samples. Lastly, Our model has a novel snippet ranking function that uses snippet prediction to obtain an alignment score and linearly combines it with the matching prediction score.

4.1 Knowledge Transfer across Tasks

We build our model upon BERT (Devlin et al., 2019) to jointly optimize on the RC and DR tasks. Suppose Θ has the BERT encoder parameters. When we apply domain knowledge transfer, which will be introduced in the following section, we initialize it with the parameters $\Theta^{(aux)}$ trained on the auxiliary domain; when we do not apply the transfer, we initialize it with the original pre-trained BERT parameters. We have two multi-layer perceptron (MLP) classifiers for the two tasks, whose parameters are denoted by θ_{RC} and θ_{DR}, respectively. Both classifiers are randomly initialized. More specifically, the RC classifier is to predict answer snippets, and the DR classifier is to predict

---

[1]Since technical domain RC is extremely difficult, we also evaluate performance on top-5 predictions in our experiments.

[2]Elasticsearch – https://www.elastic.co/elasticsearch/

[3]In our work, auxiliary domain QAs are from general domain QAs, so we use these two words interchangeably.

document matching. The joint loss is as follows:

$$\begin{aligned} \mathcal{L}^{(aux)} &= \mathcal{L}_{RC}(\Theta^{(aux)}, \theta_{RC}^{(aux)}) \\ &+ \lambda^{(aux)} \cdot \mathcal{L}_{DR}(\Theta^{(aux)}, \theta_{DR}^{(aux)}), \end{aligned} \quad (1)$$

where λ is a hyper-parameter for the weight of the DR task over RC task.

Calculate adjustment factor As shown in Eq.(1), the weights between two training objectives are only adjusted by a pre-determined hyper-parameter λ. However, for different samples in the dataset, the difficulty of learning snippet prediction and matching prediction is different. The weight of two training objectives should be dynamically adjusted so that the model can focus on optimizing the more difficult task when training different data samples. Since non-factoid questions are open-ended questions that often require complex answers that are mostly sentence-level texts, positional relationships between start token and end token in answer snippets have more fluctuations than factoid answers. Therefore, we take the difference between real answer snippet and predicted snippet to measure the difficulty of snippet prediction. Intuitively, when the predicted answer snippet is significantly different from the actual answer snippet (much larger or much smaller), it indicates snippet prediction is difficult for the current data sample. So, the model should focus on optimizing the reading comprehension part. On the contrary, the model should focus on optimizing the document retrieval part. Formally, the weight-adjustable joint learning loss function is defined as:

$$\begin{aligned} \mathcal{L}^{(aux)} &= w \cdot \mathcal{L}_{RC}(\Theta^{(aux)}, \theta_{RC}^{(aux)}) \\ &+ \lambda^{(aux)} \cdot \mathcal{L}_{DR}(\Theta^{(aux)}, \theta_{DR}^{(aux)}), \end{aligned} \quad (2)$$

$$w = \exp\left(\frac{|(D_{end} - D_{start}) - (\hat{D}_{end} - \hat{D}_{start})|}{D_{end} - D_{start}}\right). \quad (3)$$

4.2 Knowledge Transfer across Domains

Besides transferring across tasks, in our framework, we employ knowledge transfer across domains. We identify a dataset from an auxiliary domain (not a technical support domain) for technical question answering like SQuAD. We apply the multi-task learning to the auxiliary domain. The goal is to learn BERT encoder parameters $\Theta^{(aux)}$ and two MLP classifiers $\theta_{RC}^{(aux)}$ and $\theta_{DR}^{(aux)}$:

$$\begin{aligned} \mathcal{L}^{(aux)} &= \mathcal{L}_{RC}(\Theta^{(aux)}, \theta_{RC}^{(aux)}) \\ &+ \lambda^{(aux)} \cdot \mathcal{L}_{DR}(\Theta^{(aux)}, \theta_{DR}^{(aux)}), \end{aligned} \quad (4)$$
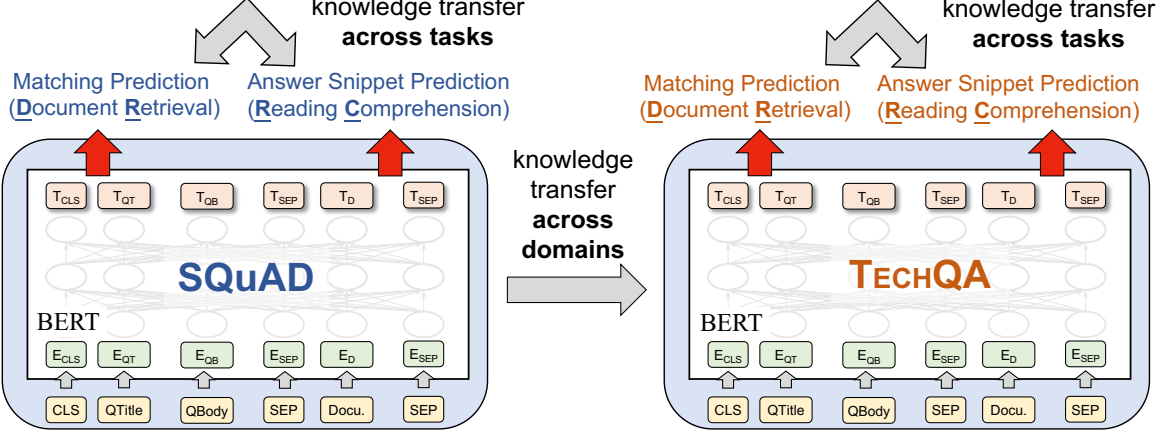
180

Figure 2: Our framework performs knowledge transfer across tasks and domains. It explores the mutual enhancement between the snippet prediction (reading comprehension) and matching prediction (document retrieval), applying multi-task learning to the BERT models on both auxiliary domain (SQuAD) and target domain (TechQA).

Here the encoder is initialized by the original pre-trained BERT parameters. We will initialize the BERT encoder in the target domain Θ with $\Theta^{(aux)}$ (used in TransTD-Mean and TransTD-CLS). When $\lambda^{(aux)} = 0$, we apply the single RC task on the auxiliary domain (used in TransTD-single).

4.3 Framework Components

Question and Document Encoder   Given a pair of question Q and document D, we first build a concatenation by $[[CLS], Q, [SEP], D, [SEP]]$, where $[CLS]$ stands for a classification token and $[SEP]$ separates components in the sequence. The BERT_Θ encoder generates contextualized representations of every token X in the input sequence q, which is denoted by $\text{BERT}_\Theta(q)[X] \in \mathbb{R}^d$, where $d = 1024$. So we have a matrix of token representations $\mathbf{H} \in \mathbb{R}^{m \times d}$, where $H(k) = \text{BERT}_\Theta(q)[q[k]]$ (k is the index of the token).

Reader MLP   This classifier reads the representation matrix $\mathbf{H}$ and computes the score of each token being the start token in the answer snippet $\mathbf{p}_{start} \in \mathbb{R}^m$ and the score of each token being the end token $\mathbf{p}_{end} \in \mathbb{R}^m$.

$$\mathbf{p}_{start} = \mathbf{w}_{start} \cdot \mathbf{H}^\mathrm{T}, \ \mathbf{p}_{end} = \mathbf{w}_{end} \cdot \mathbf{H}^\mathrm{T}, \quad (5)$$

where $\mathbf{w}_{start}, \mathbf{w}_{end} \in \mathbb{R}^d$ are trainable parameters. We have the snippet $S_{RC} = (\hat{D}_{start}, \hat{D}_{end})$ as

$$\hat{D}_{start} = \text{argmax}_{k \in \{1, \dots, m\}} p_{start}[k], \quad (6)$$

$$\hat{D}_{end} = \text{argmax}_{k \in \{1, \dots, m\}} p_{end}[k]. \quad (7)$$

Matching MLP   Suppose we have the representation of the sequence q. It can be denoted by

$\mathbf{h} \in \mathbb{R}^d$. The classifier is to predict whether the question Q and document D are aligned, which is a binary variable projected from $\mathbf{h}$:

$$p_{DR} = \sigma(\mathbf{w}_{DR} \cdot \mathbf{h}), \quad (8)$$

where σ is the sigmoid function and $\mathbf{w}_{DR} \in \mathbb{R}^d$ are trainable parameters. We have two options to produce $\mathbf{h}$ from the input sequence q. The first option is to apply mean pooling to the representations of all tokens (used in TransTD-Mean):

$$\mathbf{h} = \text{MEAN}(\{\text{BERT}_\Theta(q)[X] | X \in q\}). \quad (9)$$

The second option is to use the classification token $[CLS]$ (used in TransTD-CLS):

$$\mathbf{h} = \text{BERT}_\Theta(q)[CLS]. \quad (10)$$

Joint Inference   The reading MLP takes question and document pairs and predicts a reading score,

$$\begin{aligned} S_{reader} &= (p_{start}[D_s] + p_{end}[D_e]) \\ &\quad -(p_{start}[0] + p_{end}[0]). \end{aligned} \quad (11)$$

where $p_{(\cdot)}[0]$ denotes the probability of taking first token of the sequence as the start position or end position of the snippet. The joint ranking score of a (Q, D) pair is a linear combination of reading score and matching score,

$$S = \alpha \cdot p_{DR} + (1 - \alpha) \cdot S_{reader}. \quad (12)$$

It should be noted that different from previous work that only leverages the first term in reading score, i.e.,

Table 1: Statistics of TechQA. The test set is not publicly available, only allowing people to submit models for evaluation. The length of *TechNotes* is much bigger than that of question and answer texts.

	#Ques. (answerable/non-ans.)	#TechNotes	Len-Ques.	Len-Ans.	Len-Notes
Train	600 (450 / 150)	30,000	52.1±31.6	48.1±38.7	433.9±320.6
Dev.	310 (160 / 150)	15,500	53.1±30.4	41.2±27.7	449.1±351.2
Test	490	24,500	-	-	-

Table 2: Ablation study on knowledge transfer across tasks and across domains on TechQA. TransTD transfers knowledge across both tasks and domains, and TransTD$^+$ is further improved by the adjustable weight.

Methods	Adjustable		Source task(s)	Target task(s)	Reading Comprehension			Document Retrieval		
					Ma-F1	HA-F1@1	HA-F1@5	MRR	R@1	R@5
BERT$_{DR}$	-	✗	-	DR	-	-	-	55.80	45.58	58.23
BERT$_{RC}$	-	✗	-	RC	52.49	24.92	37.26	51.20	48.13	56.25
TransD	-	✗	RC	DR	-	-	-	60.63	58.13	64.38
	-	✗	RC	RC	55.31	34.69	50.52	64.60	60.63	68.23
TransT	CLS	✗	-	RC+DR	53.43	26.83	38.50	51.19	46.88	56.88
	Mean	✗	-	RC+DR	52.30	26.28	41.50	52.68	47.50	59.35
TransTD	CLS	✗	RC+DR	RC+DR	56.43	39.12	52.30	66.79	64.38	70.63
	Mean	✗	RC+DR	RC+DR	56.88	37.96	49.83	67.55	**67.50**	69.38
TransTD$^+$	CLS	✔	RC+DR	RC+DR	56.66	38.33	50.95	67.80	65.00	72.50
	Mean	✔	RC+DR	RC+DR	**58.58**	**40.28**	**52.57**	**67.98**	66.88	**73.13**

Table 3: TransTD outperforms two-stage retrieve-then-read methods that retrieve document based on semantic alignment. k is the number of retrieved documents.

Method	Setting	Ma-F1	HA-F1@1	R@1
BERTserini (Yang et al., 2019)	k=1	51.34	15.23	30.00
(with BM25 as retriever)	k=5	56.60	28.31	48.75
DPR (Karpukhin et al., 2020)	k=1	53.22	15.57	26.25
(w/o pre-trained retriever)	k=5	56.47	30.40	47.50
DPR (Karpukhin et al., 2020)	k=1	54.82	19.46	30.63
(with pre-trained retriever)	k=5	58.56	33.03	53.13
TransTD-Mean$^+$ (Ours, S$_{with}$)	-	**58.58**	**40.28**	**66.88**

Table 4: Our proposed snippet ranking function can bring additional improvements. Using $(p_s[0] + p_e[0])$ reflects the degree of misalignment between Q and D.

Snippet ranking method	Ma-F1	HA-F1@1	R@1
MP-BERT (Wang et al., 2019) ($S_{MP\text{-}BERT} = p_{DR} \cdot p_s \cdot p_e$)	49.45	24.65	43.75
WKLM (Xiong et al., 2020) ($S_{BERT} = \alpha \cdot p_{DR} + p_s + p_e$)	57.82	39.71	66.25
Ours (w/o document score) ($S_{w/o} = p_s + p_e - p_s[0] - p_e[0]$)	**58.58**	**40.28**	65.00
Ours (with document score) ($S_{with} = \alpha \cdot p_{DR} + S_{w/o}$)	**58.58**	**40.28**	**66.88**

$S_{reader} = (p_{start}[D_s] + p_{end}[D_e])$ (Xiong et al., 2020; Qu et al., 2020), our added second term improved inference performance. This is because during the training time, the span label of a document that does not contain an answer is set to $(0, 0)$, and such negative documents are the majority. Therefore, $(p_{start}[0] + p_{end}[0])$ reflects the probability that Q and D is not aligned. See Table 4 for experimental comparisons.

5 Experiments

5.1 TechQA Dataset

The TechQA dataset (Castelli et al., 2020) contains actual questions posed by users on the IBM DeveloperWorks forums. TechQA is designed for machine reading comprehension tasks, Each question is associated with a candidate list of 50 *Technotes* obtained by issuing a query on the search engine Elasticsearch[4]. A question is answerable if an answer snippet exists in the 50 *Technotes*, or is unanswerable otherwise. Data statistics are given in Table 1. In TechQA, the training set has 600 questions in which 450 questions are answerable; the validation set has 310 questions in which 160 questions are answerable; the test set has 490 questions. The Technotes are usually of greater length than question and answer texts.
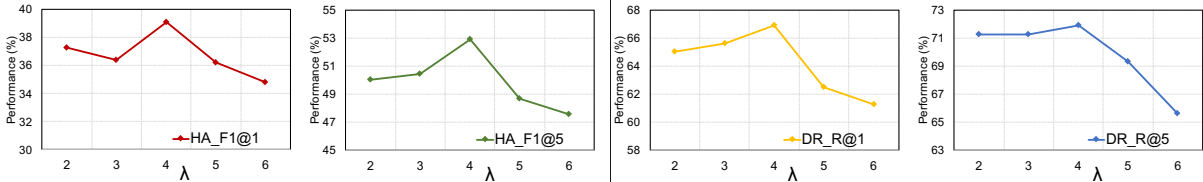
---

[4]https://www.elastic.co/elasticsearch/

Figure 3: λ is the weight of the DR task loss over the RC task loss. When $\lambda = 4.0$, TransTD achieves the best performance for both RC (left two) and DR (right two) tasks.
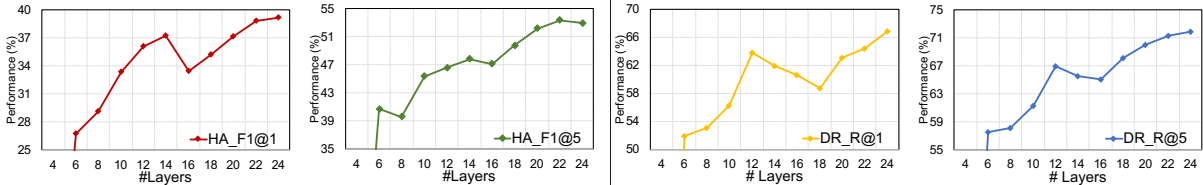

Figure 4: The more layers being fine-tuned in the target domain, the better performance we can have. However, it shows the pattern but not always true in the middle of the range.

5.2 Evaluation methods

The accuracy of the extracted snippets is evaluated by Ma-F1[5] and HA_F1@K. Ma-F1 is the macro average of the F1 scores computed on the first of the K answers provided by the system for each given question:

$$\text{Ma-F1} = \frac{\sum_{i=1}^{K} \text{F1}@K}{K}, \qquad (13)$$

where F1@K computes F1 scores for top-K answer snippets, selects the maximum F1 score, and computes the macro F1 score average over all questions. HA_F1@K calculates macro F1 score average over all answerable questions. Besides, models are evaluated on retrieving and ranking document by mean reciprocal rank (MRR) and recall at K (R@K). R@K is the percentage of correct answers in top K out of all the relevant answers. MRR represents the average of the reciprocal ranks of results for a set of queries.

5.3 Ablation Study

TranT transfers knowledge across tasks on the target domain, with multi-tasks of RC and DR.
TranD transfers knowledge from source domain RC to target domain RC w/o multi-task learning.
TransTD transfers knowledge across both tasks and domains. TransTD[+] is further improved by the adjustable weight.

---

[5]To avoid confusion between F1 (used on the TechQA leaderboard) and F1@K, we use Ma-F1 instead of F1.

5.4 Experimental Analysis

5.4.1 Knowledge transfer across domains

In Table 2, the model first fine tuning on the source domain QA (SQuAD) then further fine tuning on the target domain QA (TechQA) makes superior performance than only fine tuning on the target domain QA. This indicates knowledge transfer from general domain QA is crucial for technical QA.

5.4.2 Knowledge transfer across tasks

In Table 2, transferring knowledge across tasks better capture local correspondence and global semantic relationship between the question and document. Compared with BERT$_{\text{RC}}$, TransT improves Ma-F1 by +0.94% and HA_F1@1 by +1.91%.

5.4.3 Across both tasks and domains

In Table 2, transferring knowledge across both tasks and domains further improve model performance. TransTD fine tunes on SQuAD, then further fine tunes on the TechQA with both RC and AR tasks. It performs better than TransD and TransT. TransTD[+] makes adjustable joint learning, which further brings +1.7% and +2.32% improvements on Ma-F1 and HA_F1@1 compared to TransTD.

5.4.4 Comparison with retrieve-then-read (two-stage) methods

Using semantic similarity to predict alignment between query and document in open-domain QA is an efficient and accurate method. It can be statistical-based (e.g., BM25) (Yang et al., 2019) or neural-based that can be jointly optimized with snippet prediction (Karpukhin et al., 2020; Lee et al., 2019). However, as shown in Table 3, in
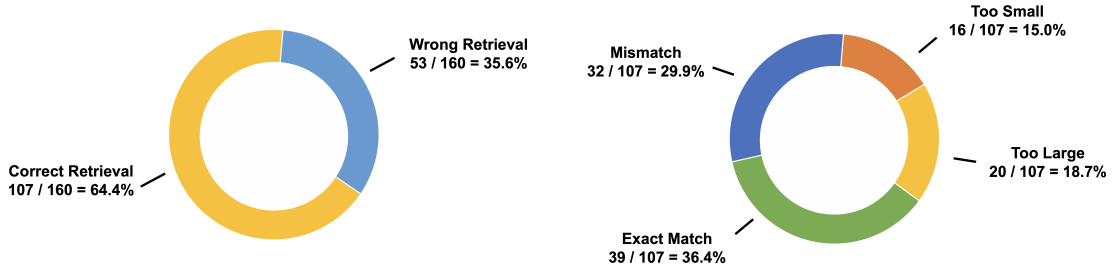
Figure 5: Error analysis. The left figure represents the proportions between correct and wrong prediction on DR. The right figure represents the proportion of RC results when the retrieval phase already predicts the correct document. (Here, "too small" means that if the prediction is $S_{RC} = (D_{start}^{(pred)}, D_{end}^{(pred)})$ and the truth is $S = (D_{start}, D_{end})$, we have $D_{start}^{(pred)} > D_{start}$ and $D_{end}^{(pred)} < D_{end}$; on the contrary, "too large" means we have $D_{start}^{(pred)} < D_{start}$ and $D_{end}^{(pred)} > D_{end}$.)

the case of the same encoder (i.e., BERT), our proposed TransTD with novel snippet ranking function can identify answers more accurately than above methods. This means that our method is more effective in the context of non-factoid QAs whose semantics of query and document are not aligned.

5.5 Parameter Analysis

Loss ratio   In Figure 3, we compare performance with loss ratio between the RC and DR tasks, λ in Eq.(1). We observe that when $\lambda = 4.0$, TransTD achieves the best performance for both RC and DR tasks. If the loss ratio becomes more than 4.0, the performance decreases significantly. This is because RC helps DR more than DR helps RC, which is consistent with results in Table 2.

Number of fine tuning layers   As shown in Figure 4, we compare performance on different numbers of fine tuning layers. Fine tuning all layers (24 layers) makes the best performance. However, the model performance and the number of fine tuning layers are not an absolute linear relationship. For example, only fine tuning 12 to 14 layers achieves better performance than having 16 or 18 layers, making a good reference for training with limited GPU memories.

5.6 Error Analysis

As shown in Figure 5, we manually categorize the predictive results of 160 answerable question instances in the development set. First of all, there are 107 (64.4%) questions that can be correctly matched with corresponding documents through the joint inference by Eq.(12), however, 53 (35.6%) questions are mismatched with the documents that do not contain desirable answers. Additionally,

among 107 correct predictions, only 39 (36.4%) of them are given with the correct answer snippet in the best matching document. Among 68 wrong predictions, 32 (47.1%) of them are mismatched with the answer span. Besides, 16 (23.5%) of them are provided with a smaller span of answer snippet than the actual span, in which the average length of answer snippet is 44 words. On the contrary, 20 (29.4%) of them are provided with a larger span of answer snippet than the actual span, in which their average length is 16 words. We observe that the TechQA dataset offers a challenging yet interesting problem, where the answers have a wide range of the number of words. Some long answers are across multiple sentences.

6 Conclusion

In this paper, we studied QA in the technical domain, which was not well investigated. Technical QA faces two unique challenges: (i) the question and answer rarely overlaps substantially (on-factoid questions) and (ii) very limited data size. To address the challenges, we propose a novel framework of deep transfer learning to effectively address TechQA across tasks and domains. To this end, we present an adjustable joint learning approach for document retrieval and reading comprehension tasks. Our experiments on the TechQA dataset demonstrates superior performance compared with non-transfer learning state-of-the-art methods.

Acknowledgements

References

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Mike McCawley, et al. 2020. The techqa dataset. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).*

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *In Proceedings of 8th International Conference for Learning Representation (ICLR).*

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).*

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37.

Yang Deng, Ying Shen, Min Yang, Yaliang Li, Nan Du, Wei Fan, and Kai Lei. 2018. Knowledge as a bridge: Improving cross-domain answer selection with external knowledge. In *Proceedings of the 27th international conference on computational linguistics*, pages 3295–3305.

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

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. On making reading comprehension more comprehensive. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 105–112.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909.*

Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2015. Social recommendation with cross-domain transferable knowledge. *IEEE transactions on knowledge and data engineering*, 27(11):3084–3097.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906.*

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.*

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. *SIGIR Conference on Research and Development in Information Retrieval.*

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5881–5885.

Lingfei Wu, Ian EH Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J Witbrock. 2018. Word mover's embedding: From word2vec to document embedding. *arXiv preprint arXiv:1811.01713.*

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *International Conference for Learning Representation (ICLR).*

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*.

Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh V Chawla, and Zhenhui Li. 2019. Graph few-shot learning via knowledge transfer. *arXiv preprint arXiv:1910.03053*.

Wenhao Yu, Lingfei Wu, Yu Deng, Ruchi Mahindru, Qingkai Zeng, Sinem Guven, and Meng Jiang. 2020a. A technical question answering system with transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wenhao Yu, Lingfei Wu, Qingkai Zeng, Yu Deng, Shu Tao, and Meng Jiang. 2020b. Crossing variational autoencoders for answer retrieval. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.