IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 21 December 2023, accepted 20 January 2024, date of publication 24 January 2024, date of current version 6 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3358287

RESEARCH ARTICLE

ReQuEST: A Small-Scale Multi-Task Model for Community Question-Answering Systems

SEYYEDE ZAHRA AFTABI, SEYYEDE MARYAM SEYYEDI, MOHAMMAD MALEKI, AND SAEED FARZI
Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran 16317-14191, Iran
Corresponding author: Saeed Farzi (saeedfarzi@kntu.ac.ir)

ABSTRACT The burgeoning popularity of community question-answering platforms as an information-seeking strategy has prompted researchers to look for ways to save response time and effort, among which question entailment recognizing, question summarizing, and question tagging are prominent. However, none has investigated the implicit relations between these tasks and the benefits their interaction could provide. In this study, ReQuEST, a novel multi-task model based on bidirectional auto-regressive transformers (BART), is introduced to recognize question entailment, summarize questions respecting given queries, and tag questions with primary topics, simultaneously. ReQuEST comprises one shared encoder representing input sequences, two half-shared decoders providing intermediate presentations, and three task-specific heads producing summaries, tags, and entailed questions. A lightweight fine-tuning technique and a weighted loss function help us learn model parameters efficiently. With roughly 187k learning parameters, ReQuEST is almost half the size of BART$_{\text{large}}$ and is two-thirds smaller than its multi-task counterparts. Empirical experiments on standard summarization datasets reveal that ReQuEST outperforms competitors on Debatepedia with a Rouge-L of 46.77 and has persuasive performance with a Rouge-L of 37.37 on MeQSum. On MediQA-RQE as a medical benchmark for entailment recognition, ReQuEST is also comparable in accuracy with state-of-the-art systems without being pre-trained on domain-specific datasets.

INDEX TERMS Community question answering systems, multi-task learning, query-focused question summarization, question entailment, tag generation.

I. INTRODUCTION

Over the last couple of decades, community question-answering (CQA) platforms have gained prominence as reliable places to acquire knowledge. Stack Overflow, Quora, and iCliniq are three examples of CQA platforms. Compared to general question-answering (QA), aiming to answer short and factoid questions automatically [1], CQA allows information seekers to post their questions in natural language with any amount of peripheral details and receive descriptive answers from human experts [2], [3], [4], [5]. Nevertheless, many questions may not be answered immediately or even remain unanswered forever. This issue may arise from the proliferation of newly arrived questions [6], too long or ambiguous questions that fail to appeal to relevant experts [7], [8], [9], improper categorization of questions or users [10], [11], [12], and the like. Hence, researchers have strived to enhance CQA platforms from various aspects and make them more efficient.

Among various techniques for improving CQA platforms, three tasks are more pivotal. The first task is recognizing question entailment (RQE) which identifies archived questions whose answers are also complete or partial answers to the input question [13]. The second task is question summarization (QS) which is an intricate task in natural language processing (NLP). QS aims to shorten the input question and generate a brief human-readable question comprising the vital information of the original one [14]. The third task is tag generation (TG) which provides a short list of keywords or phrases describing the question and its principal topics.

The associate editor coordinating the review of this manuscript and approving it for publication was Agostino Forestiero.

Two main questions may arise here: 1) Why are these tasks essential in CQA? 2) Is it possible to learn them jointly? Answering the first question requires an explanation of the applications of these tasks in CQA. Using RQE, one can redirect the questioners to entailed questions [15], generate answers automatically [13], [16], [17], or refine or supplement the input questions [18]. Besides, as questions in CQA are usually multi-sentence and complex with unnecessary details, QS enables experts to quickly discover the intent of the question [3], [19]. Further, TG helps organize questions, link similar ones, and attract pertinent experts to answer [20], [21], [22].

As for the second question we should say that most scholars have considered these tasks independent. However, a few have argued that they are interrelated and can benefit each other [3], [4], [23], [24], [25], [26], [27]. There are a number of reasons behind this argument. One reason is that although transformer-based QS models can produce fluent summaries, they do not assure the factual correctness of them [2], [28], [29]. Therefore, interacting with RQE can be fruitful since it ensures the generated summaries are logically entailed by their source texts [23], [28], [29], [30]. Another reason is that generating concise summaries requires a greater focus on the main topics of questions, which existing QS models often fail to do [27]. Consequently, while duplicate information are repeatedly given attention, essential phrases may go unnoticed [26], [27]. Therefore, as tags are good indicators for the main topics of questions, joint learning of TG and QS would provide the model with essential information in encoding layers [31]. The third reason is related to the complexities that RQE models confront, e.g., lengthy questions, lexical heterogeneity between questions, and the presence of specialized expressions. These complexities make questions hard to understand [3], [32], [33]. Thus, it could be beneficial to engage QS and TG to compress and simplify questions to abstract versions conveying pivotal information.

In this paper, we aim at boosting the RQE, QS, and TG performance by exploiting their interrelations. In particular, motivated by substantial performance gains achieved by multi-task deep neural networks over a broad range of NLP tasks [3], [34], [35], [36], [37], [38], [39], we have devised a multi-task model called ReQuEST based on encoder-decoder transformers. ReQuEST is planned to classify an input pair of questions in entailment or not-entailment classes, summarize questions by focusing on a set of thematic tags, and generate a sequence of tags stating question's main topics. It comprises one BART encoder [40] representing input sequences, two BART decoders reconstructing summary or tag sequences, and three neural network heads generating task-specific outputs. Moreover, the goal is to optimize the weighted sum of losses from all tasks while making a compromise between them. It is noteworthy that among a variety of transformers, BART[1] might be a more fitting choice given its state-of-the-art performance on several summarization benchmarks.

Joint learning of RQE and QS was also explored by Mrini et al. in [3]. They proposed a multi-task model involving one collaborative encoder and two decoders. The decoders were neither fully shared nor fully independent but gradually shared, meaning that the parameter sharing was slowly abated from the first to the last layers. There are four main differences between their effort and ReQuEST: 1) In addition to RQE and QS, ReQuEST learns another related task at the same time, which is TG. 2) The RQE decoder is omitted from ReQuEST in favor of a multi-layer neural network with fewer parameters. 3) Instead of gradually sharing the parameters between TG and QS decoders, we share only the first three layers between them and leave the remainder independent. 4) In ReQuEST, the loss coefficients in the objective function can differ depending on which component parameters are updating. However, Mrini et al. considered the coefficients constant throughout fine-tuning.

For further evaluations, a large dataset composed of question pairs, thematic tags, and entailment labels is needed. Therefore, we manipulate the well-known CQADupStack dataset [41] and introduce a new dataset named CQAD-ReQuEST in two sizes, small and large. Subsequently, several evaluation scenarios are conducted. First, the superiority of co-learning of multiple tasks over single-task learning is investigated over the small size of CQAD-ReQuEST. Second, the model sensitivity regarding to hyperparameters is analyzed. Third, the ReQuEST performance on the large size of CQAD-ReQuEST is explored. Finally, the efficiency of ReQuEST on three real-world datasets, including MeQSum [32], MediQA-RQE [42], and Debatepedia [43], is examined.

Results of the first scenario demonstrate that apart from low learning parameters, exploiting latent interactions between the related tasks brings an improvement in output quality with respect to single-task learning. The second scenario shows that ReQuEST performance does not depend highly on the initial values of hyperparameters. Finally, the experimental results in the last two scenarios confirm the generalizability of ReQuEST to other datasets, even in other domains, such as the medical domain. In a nutshell, the following two contributions are realized in this article:

1) Presenting ReQuEST, a novel compact multi-task model with few setting and learning parameters for RQE, QFQS, and TG tasks.
2) Modeling the TG task with a sequence generation method by considering it a particular type of summarization task at a relatively high granularity.

The remainder of this paper is organized into four sections. Section II covers the background and related work. The proposed method is then detailed in Section III. Section IV conveys the experimental studies, and in the end, Section V draws a conclusion and recommends research directions for the future.

[1] We use BART$_{\text{Base}}$ instead of the large variant.

II. BACKGROUND AND RELATED WORK

This section describes related work in three groups, including recognizing question entailment, text summarization, and tag recommendation.

A. RECOGNIZING QUESTION ENTAILMENT

According to the definition rendered by Abacha and Demner-Fushman [13], an entailment relation from question Q to P implies that every correct answer to P is either a complete or incomplete response to Q. In this way, RQE empowers CQA platforms to automatically answer an input question by quickly finding similar frequently asked questions (FAQs), re-ranking the corresponding answers, and selecting the top ones [42]. Therefore, it saves both the waiting time of the question asker and the effort of the answerers.

Mrini et al. [3] proposed a multi-task model using BART$_{\text{Large}}$ for joint learning of RQE and QS. On top of the shared encoder, their model contained two decoders on which a gradually soft parameter-sharing was applied. Furthermore, they evidenced an equivalence between RQE and QS in the medical domain. Correspondingly, they proposed a data augmentation approach to facilitate simultaneous multi-task learning. The empirical analysis verified that RQE helps question summarizers identify salient information from extraneous details and generate more informative summaries.

Kumar et al. [34] proposed an MT-DNN model for RQE and Natural Language Inference (NLI). They also augmented the training dataset with domain-specific data to adapt the model to specialized domains. Moreover, they used their NLI and RQE models to re-rank candidate answers in a QA task. The reported results indicated the effectiveness of their proposed data augmentation technique.

Sarrouti et al. [38] presented an MT-DNN model for relevance ranking, single sentence classification, and pairwise text classification (i.e., RQE). The basic idea was that using different but related datasets could improve the RQE performance by better capturing critical features. They also proposed an innovative data augmentation approach relying on contextualized word embedding to overcome data scarcity. They found both the RQE model and the data-augmentation scheme effective.

Zhou et al. [39] introduced an adversarial multi-task network for joint learning of RQE and QA. The model was composed of one shared BioBERT for text embedding, one shared interactive transformer for input representation, and two classifiers for QA and RQE. Additionally, a task discriminator was employed to exclude task-specific features from semantic representations through adversarial learning. Experimental analysis exhibited the outstanding performance of their proposed model.

B. TEXT SUMMARIZATION

Automatic text summarization is the process of generating an abridged form of the input text that encompasses its overall meaning. Summarization approaches can be either extractive or abstractive. In the former, summaries are produced by selecting a few tokens from the original text without modifying them. In the latter, the input text is paraphrased more coherently using intermediate conceptual representations [14], [44], [45], [46]. Abstract summaries are qualitatively close to human-written ones. Hence, generating them requires advanced NLP techniques and large-scale annotated data [46]. In recent years, research on abstractive summarization evolved from single-document to multi-document and from generic to query-focused. A query-focused abstractive summarization (QFAS) model produces a short summary that contains critical information relevant to a user-defined query [26], [47], [48].

Su et al. [47] proposed a BART-based pipeline to summarize a single document based on a query such that the generated summaries are highly-correlated with the answers provided by a QA module. They explicitly incorporated word-level answer relevance scores in the decoding process. Results showed that the use of answer relevance significantly mitigates the likelihood of copying irrelevant spans of the source text.

Laskar et al. [48] used transfer learning to overcome the need for large-scale datasets for QFAS. In the first step, they trained a BERT-SUM model using a massive repository of generic abstract summaries. In the next step, they fine-tuned it with the target dataset while considering the query relevance. Their proposed approach outperformed the state-of-the-art models.

Xu and Lapata [49] proposed a multi-task model to accomplish generic and query-focused summarization tasks together. Basically, they saw generic summarization as a specific variation of QFAS in which the query is hidden. They decomposed the learning objective into conditional language modeling and latent query modeling. Their proposed method was made up of an encoder-decoder model with two additional encoders in between. The two distinct encoders generated query-focused and query-agnostic text representations based on which the summary was generated. This idea allowed them to cope with the lack of QFAS data by utilizing generic summarization datasets.

C. TAG RECOMMENDATION

Tag recommendation is the process of generating a collection of descriptive words for a piece of text [50]. Generally, tag recommendation methods fall into two groups: personalized collaborative filtering and object-centered content-based methods. While the former ignores the input content and recommends several subjective tags based only on users' behavior history, the latter utilizes techniques such as keyword extraction, topic modeling, or text classification to determine some objective tags respecting the input content [20], [50], [51].

Lei et al. [50] proposed a classification approach using an attention-based capsule network. The capsule network encoded the spatial relationships between high-level and low-level features, then classified text entities. Meanwhile,

TABLE 1. An overview of related work.

Task	Ref.	Transformers	Datasets	Description
Recognizing question entailment	[3]	BART$_{\text{large}}$	MediQA-RQE, MeQSum, HealthCareMagic, iCliniq	*pros:* applying multi-task learning with gradual parameter sharing; utilizing pre-trained transformers; augmenting various data sources *cons:* pre-training on domain-specific data; dealing with large number of model parameters
	[34]	BERT$_{\text{base}}$	MediQA-RQE, QQP	*pros:* applying multi-task learning; augmenting generic data *cons:* incorporating domain knowledge; facing limitations in co-reference resolution and multi-hop reasoning
	[39]	BioBERT$_{\text{base}}$	BioNLP	*pros:* leveraging pre-trained transformers; excluding task-specific features *cons:* failing to understand complex syntaxes and collocation of phrases; inability to handle difficult or fuzzy relationships
	[38]	SciBERT BioBERT BlueBERT	MediQA-RQE, PubMed RCT, GARD, BioASQ, MediQA-QA, MedNLI, QQP, MedQuad, MNLI	*pros:* applying multi-task learning; utilizing pre-trained transformers; augmenting medical and open domain data based on contextual word embeddings *cons:* injecting hand-crafted features, such as question focus and question type
Text summarization	[49]	BART$_{\text{base}}$	Debatepedia, WikiCatSum, WikiRef, DUC 2006-07, CNN/Daily Mail, TD-QFS	*pros:* applicability to zero-shot settings due to inferring latent queries *cons:* relying on weak supervision, which may be noisy
	[48]	BERTSUM	Debatepedia, XSum	*pros:* addressing the scarcity of large QFAS data by utilizing generic data *cons:* pre-training on XSum (high cost)
	[47]	BART$_{\text{large}}$	Debatepedia, HotpotQA, SearchQA, DUC 2005-7, NewsQA, TriviaQA, SQuAD, NaturalQuestions	*pros:* leveraging pre-trained transformers; two-stage finetuning *cons:* relying on the performance of the answer relevance prediction model
Tag recommendation	[50]	-	TPA, AG	*pros:* capturing high-level features *cons:* witnessing performance sensitivity to hyperparameters; having limitations in capturing long-range dependencies
	[20]	-	Zhihu, Weibo	*pros:* ability to generate unseen tags; capturing semantic relations between tags *cons:* risk of generating meaningless tags; experiencing difficulties in training with randomly ordered tag sequences
	[52]	ALBERT, BERT, BERTOverflow ,CodeBERT, RoBERTa	A snapshot of Stack Overflow data dumps	*pros:* incorporating code snippets for richer representations; using pre-trained transformers *cons:* limited adaptability to emerging topics; constrained by a predefined set of tags
	[53]	RoBERTa$_{\text{base}}$	StackExchange data dumps	*pros:* predicting popular tags as well as generating unseen tags *cons:* performance dependency to vocabulary size

the attention mechanism helped distill the text's central information. Nonetheless, classification-based models often disregard tags relations and treat them as distinct categories. Besides, due to a fixed number of categories, they cannot handle dynamic tags, e.g., emerging topics.

Shi et al. [20] presented a sequence-to-sequence model to generate tags. It comprised an LSTM-based encoder to capture sequential dependencies within input text and an attention-based decoder to learn global semantic relations. Generative models can predict tags without previously seeing them in the training set, though some generated tags might be meaningless.

He et al. [52] introduced a multi-label classification method for recommending Stack Overflow tags. They utilized pre-trained language models to derive feature vectors from the question's title, description, and code snippets. These vectors were then combined to form a unified representation of the post. In their evaluation of five models, CodeBERT outperformed both a CNN-based approach and Post2Vec.

Pal et al. [53] conducted a thorough analysis of user tagging behavior across 17 StackExchange communities. They devised a transformer-based tag prediction model using a mask-filling approach with dual heads: one for predicting popular tags and another for generating finer-grained tags from user text. The model demonstrated superior performance compared to traditional methods, as measured by the Hit@k metric. An overview of related work is provided in Table 1.

III. PROPOSED METHOD

ReQuEST, a transformer-based multi-task model, jointly learns to **re**cognize **qu**estion **e**ntailment, **s**ummarize questions based on user-defined queries, and generate **t**ags. Good results of BART on abstractive summarization benchmarks compared to other encoder-decoder models motivated us to use it as ReQuEST backbone. Therefore, as depicted in Fig. 1, ReQuEST is made up of one BART-based shared encoder, two partially shared decoders, and three task-specific heads.

As shown in Fig. 1, the RQE task comprises an encoder and a multilayer neural network. It does not require any decoder since they are usually employed for text generation, which is of no significance in RQE. Besides, eliminating the decoder causes a substantial decrease in RQE parameters without compromising its performance. Fig. 1 also exhibits the QFQS task, which contains the same encoder accompanied by a decoder component and a linear head. It is expected that sharing the encoder will result in better text representations through multi-task learning. The third task illustrated in Fig. 1 is TG, whose components are identical to QFQS except for the last decoder layers and the task-specific head. Indeed, aside from the shared encoder, TG uses the first ℓ_1 layers of the decoder together with QFQS, which is hoped to be a win-win partnership. On the one hand, TG is anticipated to make the shared decoder layers focus more on keywords and keep the generated sequence as short as possible. On the other hand, QFQS is likely to make the shared decoder layers attend more to the meanings. In the meantime, the independent decoder layers would allow them to realize their task-specific objectives individually.

ReQuEST parameters are tuned by minimizing the weighted addition of cross-entropy losses of all tasks. However, due to the prohibitive cost of fully fine-tuning the pre-trained language models for downstream tasks, the number of learning parameters is decreased using lightweight fine-tuning. More specifically, there are some recently developed techniques, such as adapter-tuning [54], in-context learning [55], prefix-tuning [56], and lightweight fine-tuning, which allow researchers to update only a small subset of model parameters without degrading performance. This study follows the last one and freezes the first three layers of the shared encoder during fine-tuning. Yet, one could freeze any number of encoder or decoder layers. The number of parameters per task is itemized in Table 2.

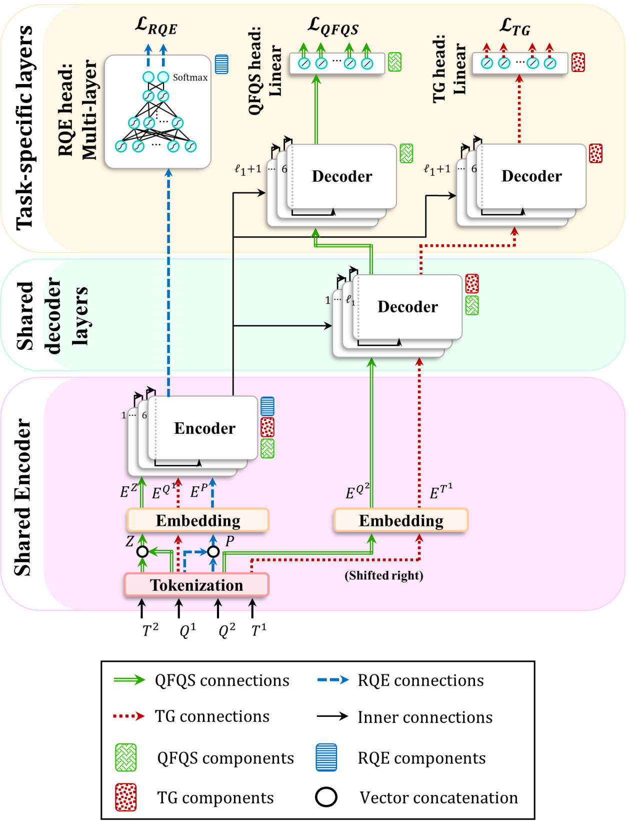


FIGURE 1. The proposed multi-task method (ReQuEST).

TABLE 2. The number of trainable parameters in each task.

Component	Number of trainable parameters
BART$_{\text{Large}}$	≈ 400 M
BART$_{\text{Base}}$	≈ 140 M
RQE	≈ 21 M
QFQS	≈ 117 M
TG	≈ 117 M
RQE + QFQS + TG (before sharing)	≈ 256 M
ReQuEST	**≈ 187 M**

A. FORWARD PROCEDURE

*RQE:*Suppose $< Q^1, Q^2 >$ is a pair of questions that, once taken by RQE, it should determine whether or not the correct answer to Q^2 can be a partial or complete correct answer to Q^1. To this end, RQE pursues two main steps: input representation and binary classification.

The former includes sequence tokenization, concatenation, and representation. Precisely, Q^1 and Q^2 are first tokenized as formulated by (1). Next, a single token sequence called P is constructed by concatenating tokenized sequences and adding special tokens, i.e., <bos> and <eos>.[2] Then, the shared encoder provides a contextualized representation of P, defined by (2).

$$Q^1 : \{t_i^1\}_{i=1}^{|Q^1|}, Q^2 : \{t_i^2\}_{i=1}^{|Q^2|} \tag{1}$$

$$\mathbb{H}^P = \{\mathbb{h}_i\}_{i=1}^{|Q^1|+|Q^2|+4} \tag{2}$$

here, $|Q^1|$ and $|Q^2|$ are the number of tokens in Q^1 and Q^2, respectively.

In the latter, the RQE head predicts the label given the final representation of the $<$ bos $>$ token in P, i.e., $\mathbb{h}_0$. In other words, a multi-layer fully-connected network, equipped with the *Tanh* activation function and a *Softmax* layer on top, learns the complex nonlinear mapping between the question pair embedding and the target label $l \in \{0, 1\}$. Accordingly, RQE aims at minimizing the binary cross-entropy loss, calculated

[2] P is constructed as follows, $< bos > t_1^{Q^1} t_2^{Q^1} \cdots t_{|Q^1|}^{Q^1} < eos >< eos > t_1^{Q^2} t_2^{Q^2} \cdots t_{|Q^2|}^{Q^2} < eos >$. Hence, the number of tokens in P would be $|Q^1| + |Q^2| + 4$.

by (3).

$$\mathcal{L}_{RQE} = -\frac{1}{M}\sum_{i=1}^{M} l_i \log \hat{l}_i + (1-l_i)\, log\left(1-\hat{l}_i\right) \tag{3}$$

where M is the total number of question pairs, and $\hat{l}_i$ is the predicted label for the ith pair.

QFQS: Let Q^1 be the submitted question in a CQA platform. Upon submission, a condensed version of Q^1 must be generated by QFQS in which contextual information is included at both syntactic and semantic levels. Yet, preserving coherence and integrity around the main topics remains a challenge. One possible solution is to condition the summarization task on several specific topics. Topics could also be substituted by tags because tags in CQA platforms are usually used to categorize questions topically. Thence, suppose that a set of m tags is also specified for each question, with m typically ranging from 0 to 5.

The proposed QFQS model is planned to summarize questions in five steps. In the first step, tags are joined together by whitespace characters to build a single sequence called T. As soon as Q^1 and T are separately tokenized, a new token sequence called Z is built in the second step by appending them together with special tokens.[3] In the third step, Z is fed to the shared encoder to compute the contextualized word representations, denoted by $\mathbb{H}^Z$ in (4).

$$\mathbb{H}^Z = \{\mathbb{h}_i\}_{i=1}^{|Q^1|+|T|+4} \qquad (\mathbb{h}_i \in \mathbb{R}^{1\times d}) \tag{4}$$

where $|T|$ denotes the number of tokens in the tag sequence.

During the fourth step, the decoding process is carried out upon receipt of the shared encoder outputs. In this case, target summaries must also be provided to the decoder for training purposes. To this end, every question Q^2 can be regarded as a promising target summary for Q^1 if and only if Q^1 is longer than Q^2 and entails it. This idea comes from the study of Mrini et al. in [3]. In the final step, d-dimensional vectors from the last decoder layer are projected into $\mathcal{v}$-dimensional spaces using the QFQS head, where $\mathcal{v}$ indicates the vocabulary size. In other words, for every position j in the sequence, the goodness of each token $t \in$ Vocab is measured. The more the goodness score is, the more that token is preferred to be placed in that position. As a point of note, during training, the cross-entropy loss estimated based on (5) is adopted.

$$\mathcal{L}_{QFQS} = -\frac{1}{M}\sum\nolimits_{i=1}^{M}\sum\nolimits_{j=1}^{|S_i|} t_j^{S_i} log t_j^{\hat{S}_i} \tag{5}$$

where the reference and generated summaries are respectively shown by S_i and $\hat{S}_i$. Furthermore, $t_j^{S_i}$ and $t_j^{\hat{S}_i}$ indicate the jth token in the reference summary and generated summary.

TG: With Q^1 as the input question, the TG model should automatically provide a short list of tags describing its key topics. The proposed TG model is developed based on sequence-to-sequence methods because, compared to classification approaches, they can generate even unseen tags. As such, the sequence of tags is treated like a sentence to be generated. Moreover, a sequence of tags can also be viewed as a summary at a relatively coarse granularity, which has fewer details, limited length, no sequential dependencies, and no grammar constraints. Consequently, an encoder-decoder architecture analogous to QFQS is needed for the TG task.

Based on ReQuEST framework depicted in Fig. 1, TG also takes advantage of the shared encoder to attain contextualized word representations of Q^1. The shared encoder outputs are then fed to the decoder and go through its layers consecutively. Based on the above discussion on similarities and differences between TG and QFQS, we propose to share the first decoder layers between them and leave the rest free. Hence, they could benefit each other while retaining their independence to achieve their objectives. Indeed, once reaching the ℓ_1th layer, the path is separated from QFQS towards $(6 - \ell_1)$ exclusive layers. The outputs of the last exclusive layer are finally passed to the TG head to reproduce tag sequences. It is of note that all trainable parameters in TG are optimized using the cross-entropy loss computed by (6).

$$\mathcal{L}_{TG} = -\frac{1}{M}\sum\nolimits_{i=1}^{M}\sum\nolimits_{j=1}^{|T_i|} t_j^{T_i} log t_j^{\hat{T}_i} \tag{6}$$

where $t_j^{T_i}$ implies the jth token in the reference tag sequence and $t_j^{\hat{T}_i}$ designates the corresponding generated token.

B. TRAINING PROCEDURE

The main advantage of multi-task learning is that multiple related tasks can implicitly benefit each other by participating in the regulation of shared parameters. Ergo, as formulated by (7), multi-task models usually employ a linear weighted sum of losses as the total loss for optimization.

$$\mathcal{L}_{Total} = \sum\nolimits_{i\in\{tasks\}} \alpha_i \mathcal{L}_i \tag{7}$$

where $\mathcal{L}_i$ is the loss of task i, and α_i is its corresponding coefficient. However, this approach may pose a problem because of identical loss coefficients throughout all layers. To clarify, all learnable parameters contributing to task i, whether in task-specific or shared layers, are updated based on a single coefficient of $\mathcal{L}_i$. Consequently, using large coefficients may cause the model to overfit quickly, and using small ones may delay the convergence. In this study, the loss coefficients are exclusively determined for each component, and thus, the gradients have to be recalculated to update each component. In particular, the weighted sum of losses of all tasks, computed based on (8), is utilized for updating all trainable parameters in the shared encoder component. Further, as calculated by (9), shared decoder layers are optimized by incorporating losses of TG and QFQS. Lastly, task-specific parameters are tuned by back-propagating their corresponding losses.

$$\mathcal{L}_{Enc} = \alpha\mathcal{L}_{RQE} + \beta\mathcal{L}_{QFQS} + \gamma\mathcal{L}_{TG} \tag{8}$$

[3] Z is constructed as follows, $< bos > t_1^{Q^1} t_2^{Q^1} \cdots t_{|Q^1|}^{Q^1} < eos >< eos > t_1^T t_2^T \cdots t_{|T|}^T < eos >$.

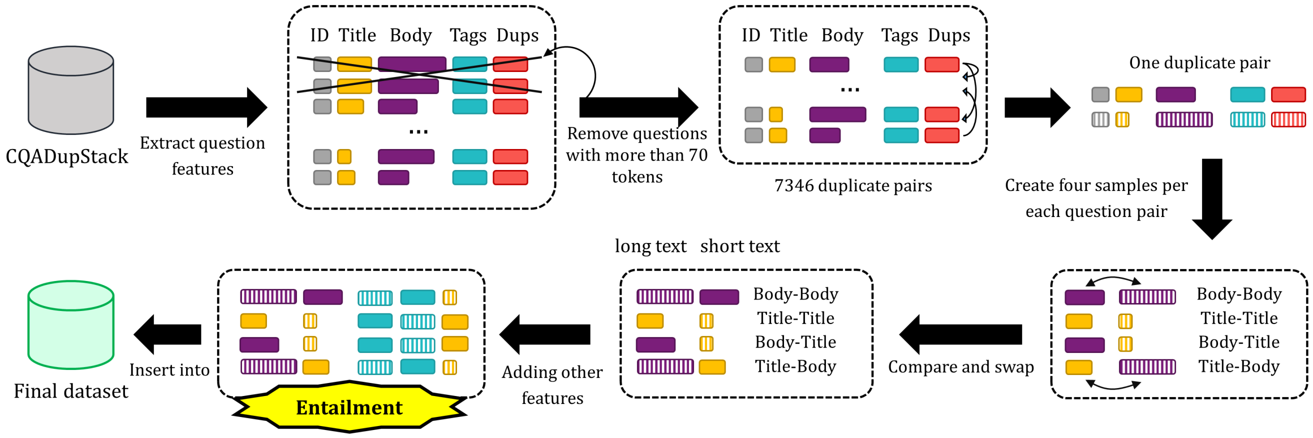


FIGURE 2. **CQAD-ReQuEST data formation steps.**

TABLE 3. **Some statistics about the datasets.**

Dataset	# samples⅄		# tokens in Q^1			# tokens in Q^2			# tokens in T^1			# tokens in T^2			Train/Test percentage
	+	–	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	
MediQA-RQE	4769	4048	6	196	26.88	5	53	15.99	-	-	-	-	-	-	~97% / ~3% ✧
Debatepedia	13000	0	6	169	74.42	5	35	14.77	-	-	-	5	29	14.99	~92% / ~8% ✧
MeQSum	1000	0	7	417	66.40	6	43	14.37	-	-	-	-	-	-	50% / 50% ✷
CQAD-ReQuEST	**29357**	**29349**	**5**	**129**	**36.98**	**4**	**90**	**16.90**	**3**	**22**	**6.52**	**3**	**22**	**6.43**	**~96% / ~4%** ✷

⅄ The sign (+) denotes the entailed class while (–) means the class of not-entailed samples
✧ Train and test splits have been provided by the dataset publisher
✷ The dataset has been randomly split

$$\mathcal{L}_{Sh\_Dec} = \rho\mathcal{L}_{QFQS} + \tau\mathcal{L}_{TG} \tag{9}$$

The coefficients α, β, γ, ρ, and τ are five real-valued hyperparameters whose best combination should be estimated through trial and error.

IV. EXPERIMENTAL STUDY

There are four critical research questions (RQ) that should be thoroughly explored:

- **RQ1:** How is the efficiency of simultaneous learning of multiple tasks compared to single-task learning?
- **RQ2:** How sensitive is ReQuEST to changes in coefficients? Which combination of coefficients yields the highest performance in all tasks?
- **RQ3:** How well does ReQuEST perform on the whole CQAD-ReQuEST dataset?
- **RQ4:** How efficient is ReQuEST on other datasets, whether open domain or restricted domain?

Further in this section, datasets, model configuration, and evaluation criteria are expounded on. Thereafter, respecting four research questions, experimental analysis is performed in four distinct scenarios: 1) comparing simultaneous multi-task learning with single-task learning in terms of F1-score and Accuracy for the RQE task, and Rouge-L and BERTScore for QFQS and TG tasks, 2) analyzing the sensitivity of ReQuEST performance to changes in coefficients, 3) analyzing ReQuEST performance on the large size of CQAD-ReQuEST dataset, and 4) investigating the efficiency of ReQuEST in comparison with some recently proposed models over three well-known datasets.

A. DATA

The statistics of three public datasets on which the proposed method is appraised are reported in Table 3. What emerges from Table 3 is that these datasets are not only small in size but also inappropriate for simultaneous multi-task learning on RQE, QFQS, and TG. To address these issues, a new dataset, hereafter called CQAD-ReQuEST, is also developed by extracting the needed information from an existing dataset named CQADupStack [41].

MediQA-RQE[4] [42] is a medical dataset including pairs of consumer health questions and frequently asked questions labeled manually by medical experts. Note that the test set greatly varies from the training set, and thus, many previous studies have described their test results as unfavorable despite achieving good results on validation data [8], [36], [57], [58].

Debatepedia[5] [43] is the earliest large dataset for query-focused abstractive summarization. It is comprised of

[4] https://github.com/abachaa/MEDIQA2019/tree/master/MEDIQA_Task2_RQE

[5] https://github.com/PrekshaNema25/DiverstiyBasedAttention Mechanism/tree/master/data

TABLE 4. The values of setting parameters.

Parameter	Value
Embedding size	768
Dropout rate	0.1
Label smoothing	0.1
Warmup*	0
Total number of epochs*	10
Batch size*	8
Frozen layers of the shared encoder*	The first 3 layers
Frozen layers of the QFQS decoder	None
Frozen layers of the TG decoder	None
Shared decoder layers (ℓ_1) *	The first 3 layers
Number of neurons in each fully-connected layer of the RQE head*	48, 24, 2
Number of beams (in Beam search)	4^a, 1^b
Maximum length of generated sequences	35^a, 15^b
Minimum length of generated sequences	8^a, 3^b
No-repeat n-gram size	3^{ab}

a: In the query-focused question summarization task
b: In the tag generation task
*: Parameters tuned by trial-and-error.

documents, queries, and summaries, extracted from a collection of pro and con quotations about debate issues. It is worth noting that queries are formal natural language questions.

MeQSum [6] [32], consists of 1,000 questions collected from the U.S. National Library of Medicine which are annotated by summaries written by three medical experts. In addition to the text of the message, for most questions, the subject is also specified.

CQAD-ReQuEST[7] is a dataset provided in this paper via modifying CQADupStack,[8] a public benchmark dataset for community question answering. It is well-suited for joint learning of RQE, QFQS, and TG tasks. In other words, it enables us to optimize all the tasks concurrently, rather than successively training them with distinct datasets.

As shown in Fig 2, CQAD-ReQuEST is constructed through six steps. In the initial step, essential features, including ID, body, title, tags, and duplicate question IDs, are extracted for each question. The second step employs a filtering mechanism to identify questions with fewer than 70 tokens, resulting in a refined collection of 202,304 questions with 7,346 duplicate pairs. The third step is to create four samples per each question pair, Q^1 and Q^2. This involves coupling their bodies, associating their titles, and pairing the body of Q^1 with the title of Q^2, and vice versa. These samples are then integrated into a new empty dataset. The fourth step entails comparing the first sequence of each sample against the second one, with a swap made if necessary to ensure that the first sequence always stands as the longest. Henceforth, they are referred to as the ''long'' and ''short'' text, respectively. The fifth step enriches each sample by appending the tags of both texts and the title of the long text. Finally, in the last step, all samples are uniformly labeled as 1, denoting their membership in the entailment class. The whole procedure is then similarly applied to non-duplicate questions, albeit with labels set to 0.

During simultaneous multi-task learning, the long and short texts and their label are utilized for the RQE task. Meanwhile, TG parameters are learned using long texts as input sequences and long text tags as target sequences. For QFQS purposes, if the label is 1, the long and short texts are regarded as the input and target sequences; otherwise, long texts and their titles are utilized. This study also introduces CQAD-ReQuEST$_{\text{small}}$, a downsized version with 7,992 training samples and 1,999 test samples, mirroring the test set in the larger dataset.

B. CONFIGURATION AND SETTING PARAMETERS

We built the proposed method in Python using Huggingface Transformers library and executed it on a Google Colab environment with a Tesla T4 GPU and 12.68 GB RAM. Table 4 outlines the hyperparameters and their initial values used in the experiments. It is of note that the maximum and minimum lengths of generated sequences are set based on the statistics in Table 3.

C. EVALUATION METRICS

We evaluate the performance of the QFQS and TG models using the standard Rouge metric (Recall-Oriented Understudy for Gisting Evaluation) [59] and BERTScore [60]. Moreover, the accuracy, which measures the ratio of correctly classified samples, and the F1-score, which is the reciprocal of the average of precision and recall, are used to assess the RQE task. However, F1-score presents a better assessment than accuracy when the data is highly imbalanced. This section defines Rouge and BERTScore and argues about their reasonability.

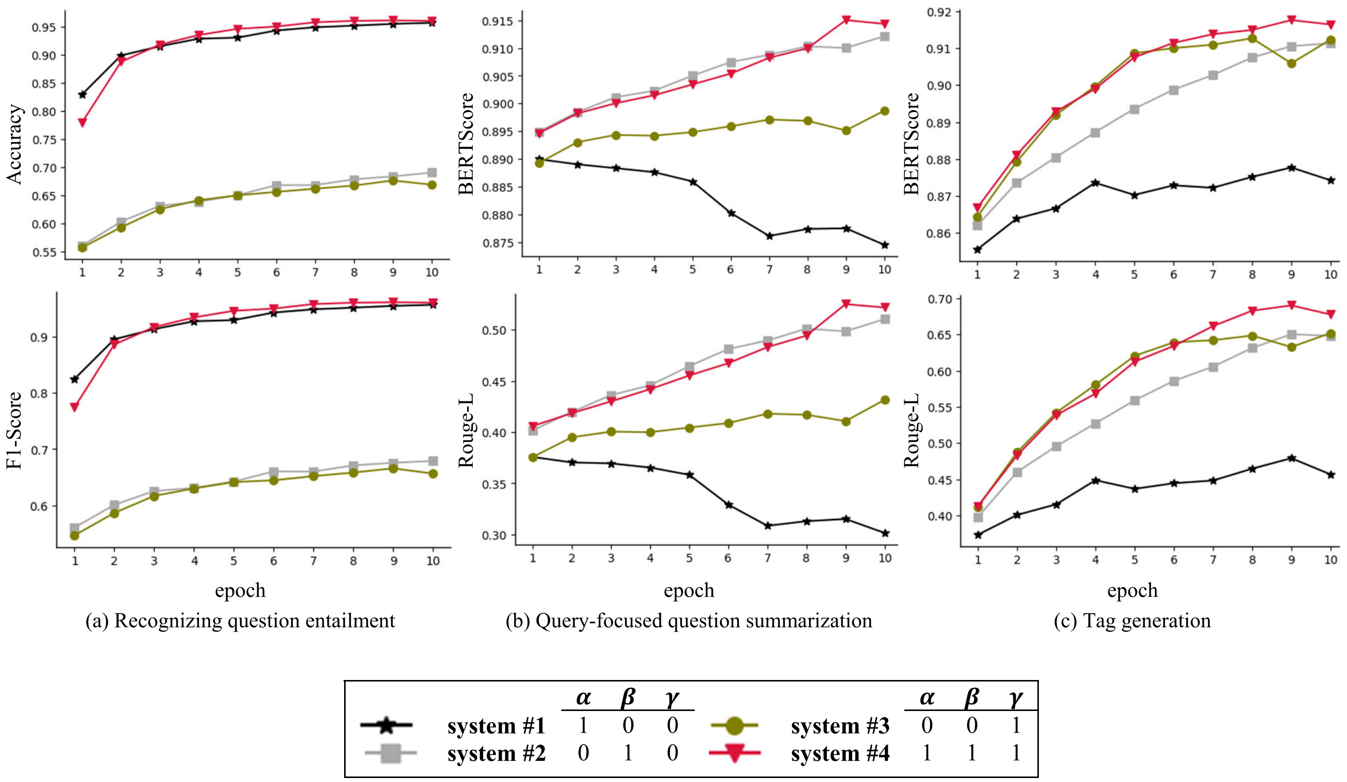
1) ROUGE

As an evaluation criterion for text summarization tasks, Rouge is the most prevalent. Equation (10) formulates the recall version of Rouge-N. Taking S as the original sequence and $\hat{S}$ as the generated one, the numerator enumerates the overlapping N-grams between them, and the denominator counts the total of N-grams in S.

$$\text{Rouge} - \text{N} = \frac{\sum_{gram_N \in S} Count_{match}\left(gram_N\right)}{\sum_{gram_N \in S} Count\left(gram_N\right)} \tag{10}$$

Rouge-N values fall between 0 and 1, with closer values to 1 indicating more lexical overlap and a better quality of the generated sequences. Although Rouge-N ensures the model captures the reference sequence's information as much as possible, it does not consider the word order when N is small. In these cases, Rouge-L is recommended in which the longest common sequence of words (LCS) that are not necessarily consecutive is regarded, as calculated by (11). In this study,

[6] https://github.com/abachaa/MeQSum
[7] Our dataset is available on GitHub: https://github.com/SZAftabi/ CQAD-ReQuEST
[8] http://nlp.cis.unimelb.edu.au/resources/cqadupstack/



(a) Recognizing question entailment

(b) Query-focused question summarization

(c) Tag generation

FIGURE 3. Comparison of model performance in the case of simultaneous multi-task learning and cases where the shared parameters are tuned based on only one task. Each plot shows a competition among four systems based on a specific evaluation criterion during the training procedure.

Rouge-1, Rouge-2, and Rouge-L are reported.

$$\mathrm{Rouge} - \mathrm{L} = \frac{LCS\left(gram_N\right)}{|S|} \tag{11}$$

2) BERTSCORE

One of the evaluation metrics for text generation tasks, introduced in 2019, is BERTScore. In contrast to Rouge metrics that measure syntactic similarity, BERTScore compares the reference sequence to the generated one semantically. According to (12), first, the contextualized word embedding of both sequences are obtained by passing them to a BERT-based transformer model (i.e., E^S and $E^{\hat{S}}$). Second, for every token in the reference sequence, the cosine similarity relative to each token in the generated sequence is calculated. Lastly, recall is computed by doing a greedy matching.

$$\mathrm{BERTScore_{Recall}} = \frac{1}{|S|} \sum\nolimits_{e_i \in E^S} \max_{\hat{e}_j \in E^{\hat{S}}} e_i^T \hat{e}_j \tag{12}$$

Here, e_i and $\hat{e}_j$, respectively, are the embedding of the ith token in S and $\hat{S}$.

D. MULTI-TASK LEARNING VS. SINGLE-TASK LEARNING

ReQuEST is able to perform three related tasks, i.e., RQE, QFQS, and TG, using shared parameters. This section addresses RQ1 (*How is the efficiency of simultaneous learning of multiple tasks compared to single-task learning?*) by comparing the performance of ReQuEST when learning all tasks simultaneously to when learning them individually. The results are shown in Fig. 3, representing each task in a separate column, with the horizontal and vertical axes indicating the iteration and the evaluation criterion, respectively.

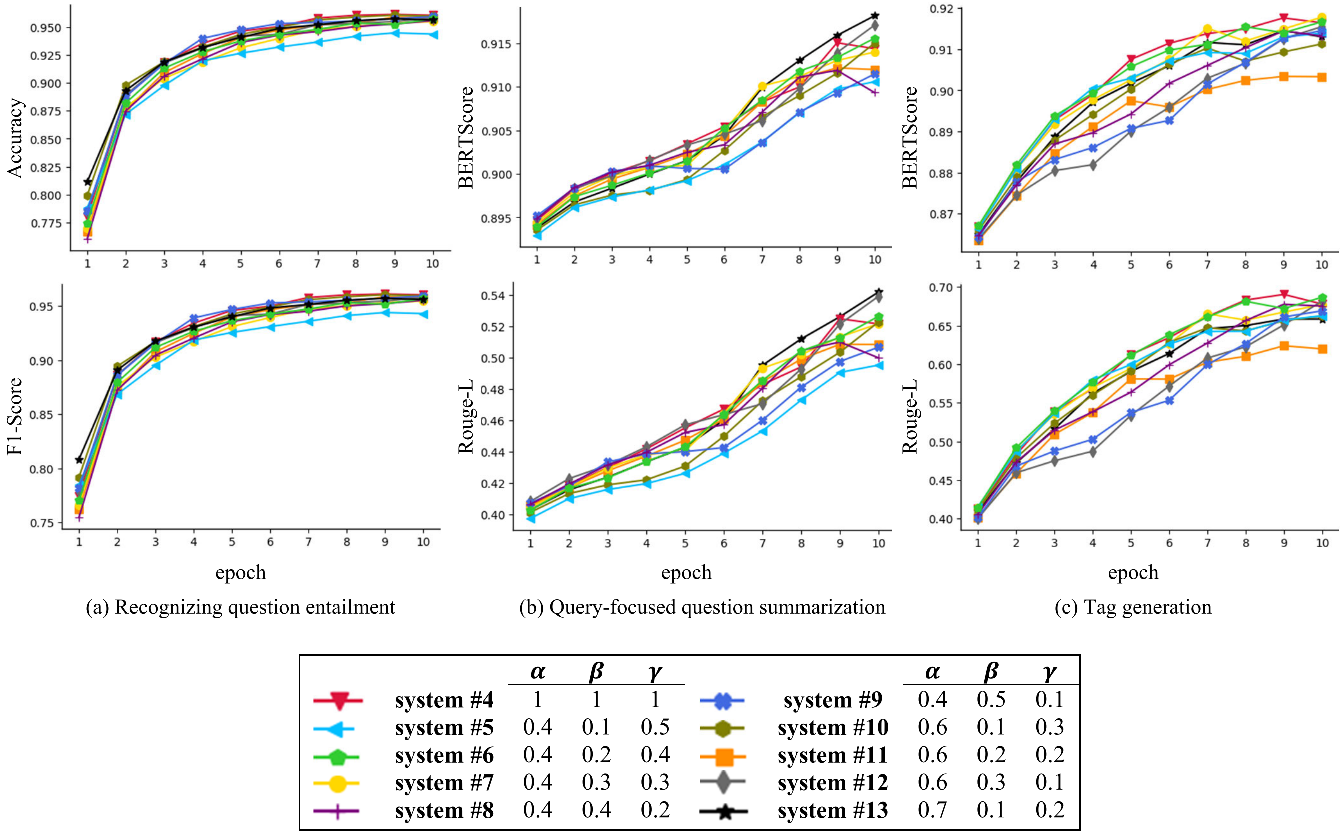
According to Fig. 3, four systems are evaluated on CQAD – ReQuEST$_{\mathrm{small}}$. In system #1, the shared encoder parameters are fine-tuned exclusively based on the loss of the RQE task. Likewise, in system #2, all parameters contributing to QFQS are updated solely based on the loss of the QFQS task, and in system #3, only the TG loss is considered for the participated parameters in the TG task. Note that, task-specific heads receive updates across all systems tailored to their individual task losses. System #4 follows the main idea of ReQuEST, that is to adjust all trainable parameters with the simultaneous contribution of all tasks. It is worth mentioning that BERTScore and Rouge-L values are recall-oriented.

From the viewpoint of RQE performance, shown in Fig. 3 (a), system #4 achieves competitive performance with system #1 in terms of F1-score and Accuracy. Nonetheless, the QFQS results depicted in Fig. 3 (b) reveal the superiority of system #4 over system #1. In particular, system #1 fails to adjust the shared encoder parameters in such a way that also improves the QFQS performance. Hence, it witnesses a decreasing trend in BERTScore and Rouge-L, while system #4 achieves the best performance in the QFQS task. A similar reasoning can also be applied to the results of the TG task

TABLE 5. The performance comparison between single-task and multi-task learning in ReQuEST on CQAD – ReQuEST$_{small}$ test data.

System	Coefficients			RQE				QFQS				TG			
	α	β	γ	Acc	F1	Re	Pr	BS	RL	R1	R2	BS	RL	R1	R2
#1	1	0	0	**0.886**	**0.885**	**0.868**	0.903	0.860	0.221	0.238	0.107	0.860	0.346	0.395	0.174
#2	0	1	0	0.678	0.664	0.632	0.699	0.875	0.287	0.309	0.157	0.876	0.438	0.484	0.281
#3	0	0	1	0.654	0.643	0.619	0.669	0.880	0.319	0.344	0.184	**0.881**	0.458	0.502	0.303
#4	1	1	1	0.878	0.875	0.847	**0.904**	**0.886**	**0.354**	**0.381**	**0.216**	0.876	**0.468**	**0.509**	**0.314**

Acc: Accuracy, F1: F1-Score, Re: Recall, Pr: Precision, BS: BERTScore, RL: Rouge-L, R1: Rouge-1, R2: Rouge-2



(a) Recognizing question entailment

(b) Query-focused question summarization

(c) Tag generation

FIGURE 4. Comparison of model performance for various compositions of coefficients. Each plot shows a competition among ten systems based on a specific evaluation criterion during the training procedure.

presented in Fig. 3 (c). In the earliest iterations, system #4 exhibits performance on par with system #3 but gradually overtakes it in the last iterations. Its supremacy is also evident from the point of view of QFQS and RQE tasks. In addition to systems #1 and #3, system #2 is also defeated by system #4 due to poor performance in RQE and TG tasks.

Overall, the observations in Fig. 3 show that ReQuEST outperforms individual models despite reducing the number of trainable parameters significantly. Further, the general attitude of most plots is increasing, underscoring the positive impact of RQE on TG and TG on QFQS through latent interactions. Table 5 also confirms this argument by reporting the evaluation results on test data. Bearing in mind that BART is pre-trained for sequence generation tasks such as text summarization, increasing iterations may cause the QFQS task to overfit. Hence, the training procedure is stopped after ten iterations in all experiments.

E. SENSITIVITY ANALYSIS

In response to RQ2 (*How sensitive is ReQuEST to changes in coefficients? Which combination of coefficients yields the highest performance in all tasks?*), we measure the model performance for different coefficients. Nevertheless, as coefficients are real numbers (i.e., $\in \mathbb{R}$), the number of possible combinations is infinite. Thus, for simplicity, we set the coefficients ρ and τ to 1 and designate the remaining coefficients, i.e., α, β, and γ in such a way that they sum to 1. A selection of the experimental results is exhibited in Fig. 4.

Fig. 4 illustrates the changes in model performance for ten systems. Each task has its own column, with plots

TABLE 6. Analysis of ReQuEST performance on test data of CQAD – ReQuEST$_{\text{small}}$.

System	Coefficients			RQE				QFQS				TG			
	α	β	γ	Acc	F1	Re	Pr	BS	RL	R1	R2	BS	RL	R1	R2
#4	1	1	1	0.878	0.875	0.847	0.904	0.886	0.354	0.381	0.216	0.876	0.468	0.509	0.314
#5	0.4	0.1	0.5	0.878	0.876	0.860	0.894	0.882	0.319	0.340	0.191	**0.880**	0.470	0.516	0.317
#6	0.4	0.2	0.4	0.880	0.876	0.836	0.919	0.886	0.347	0.371	0.213	**0.880**	0.476	0.519	0.317
#7	0.4	0.3	0.3	0.860	0.847	0.773	**0.937**	0.884	0.341	0.370	0.197	0.879	0.477	0.517	0.328
#8	0.4	0.4	0.2	0.885	0.886	0.885	0.886	0.878	0.309	0.337	0.169	0.872	0.435	0.482	0.274
#9	0.4	0.5	0.1	0.882	0.878	0.841	0.918	0.884	0.334	0.360	0.197	0.877	0.468	0.510	0.314
#10	0.6	0.1	0.3	0.885	0.882	0.857	0.909	0.886	0.355	0.383	0.218	0.879	0.479	0.519	0.328
#11	0.6	0.2	0.2	0.882	0.876	0.828	0.931	**0.889**	**0.367**	**0.393**	**0.229**	0.873	0.466	0.510	0.304
#12	0.6	0.3	0.1	0.870	0.873	0.885	0.862	0.885	0.346	0.372	0.212	0.873	0.449	0.492	0.290
#13	0.7	0.1	0.2	**0.889**	**0.890**	**0.889**	0.890	0.888	0.361	0.386	0.226	0.876	**0.493**	**0.537**	**0.340**
mean (variance)				0.879 (7e-05)	0.876 (1e-04)	0.850 (1e-03)	0.905 (5e-04)	0.885 (1e-05)	0.343 (3e-04)	0.369 (4e-04)	0.207 (3e-04)	0.877 (9e-06)	0.468 (3e-04)	0.511 (2e-04)	0.313 (4e-04)

Acc: Accuracy, F1: F1-Score, Re: Recall, Pr: Precision, BS: BERTScore, RL: Rouge-L, R1: Rouge-1, R2: Rouge-2

TABLE 7. Analysis of ReQuEST performance after training for ten iterations on CQAD-ReQuEST while having $\alpha = 0.7$, $\beta = 0.1$, and $\gamma = 0.2$.

Task / Data	RQE				QFQS				TG			
	Acc	F1	Re	Pr	BS	RL	R1	R2	BS	RL	R1	R2
Train	0.933 (-0.023)	0.931 (-0.025)	0.908 (-0.038)	0.955 (-0.011)	0.899 (-0.019)	0.422 (-0.120)	0.447 (-0.113)	0.288 (-0.149)	0.902 (-0.011)	0.603 (-0.055)	0.634 (-0.053)	0.481 (-0.070)
Test	0.900 (+0.011)	0.900 (+0.010)	0.886 (-0.003)	0.914 (+0.024)	0.885 (-0.003)	0.350 (-0.011)	0.376 (-0.010)	0.213 (-0.013)	0.898 (+0.022)	0.606 (+0.113)	0.636 (+0.099)	0.492 (+0.152)

Acc: Accuracy, F1: F1-Score, Re: Recall, Pr: Precision, BS: BERTScore, RL: Rouge-L, R1: Rouge-1, R2: Rouge-2

representing changes in corresponding evaluation metrics over time. As demonstrated in Fig. 4 (a) there is intense competition among different systems as to accuracy and F1-Score. Even so, all systems eventually converge to roughly stable F1 scores within the range [0.943, 0.960]. This phenomenon indicates that ReQuEST performance in the RQE task is relatively independent of the exact tuning of coefficients. In Fig. 4 (b), upward trends in BERTScore and Rouge-L results are evident for all systems, although the slopes differ. The final BERTScore values ranging in [0.909, 0.918] indicate that semantic similarities are well preserved in all systems. Furthermore, the broader range of Rouge-L values, i.e., [0.495, 0.542], implies that systems adhere more to semantic than lexical similarity, which makes sense in abstractive summarization. In a nutshell, Fig. 4 (b) reveals that all systems have the potential to converge to efficient results, though some do so faster, and some require more time. A similar analysis can be made for the TG task based on Rouge-L and BERTScore results exhibited in Fig. 4 (c). Indeed, not a high γ coefficient necessarily means the best proficiency in TG, but rather the alliance of the three tasks. In conclusion, the sensitivity analysis verifies that ReQuEST has stable performance under various coefficients, and thus a wide range of coefficients can lead to expedient outcomes.

Table 6 reports the final performance achieved by systems on test data per task in detail. The mean and variance of each evaluation metric over ten systems are also announced. Table 6 implies that ReQuEST can achieve acceptable results on all three of its tasks concurrently. Besides, variance values close to zero confirm its stable behavior.

F. ANALYSIS OF REQUEST PERFORMANCE ON CQAD-REQUEST

To answer RQ3 (*How well does ReQuEST perform on the whole CQAD-ReQuEST data?*), we repeat training ReQuEST using the existing 58k samples in the CQAD-ReQuEST dataset. Accordingly, a combination of coefficients must first be chosen for training, even though ReQuEST performance is almost independent of them. Based on Table 6, we pick the coefficients of system #13 (i.e., 0.7, 0.1, and 0.2) because of its superior performance in both RQE and TG tasks and achieving the second-best outcomes in QFQS. The evaluation results are presented in Table 7. Furthermore, to illustrate whether an improvement is attained, the difference between each value and the corresponding result obtained by system #13 is displayed in parentheses. It is worth mentioning that this scenario utilizes the same test data as the second scenario to make results comparable. Moreover, the number of iterations remains unchanged as our large-scale dataset makes training iterations more time-consuming.

The evidence in Table 7 supports the claim that ReQuEST performs all its tasks fairly well. It also implies that the data distribution in CQAD-ReQuEST is sufficiently homogeneous. For more justification, negative differences reported for the training data are mainly attributed to the early stopping of the training procedure. Indeed, as the dataset is large-scale and diverse, ReQuEST needs more iterations to become

TABLE 8. Comparative analysis of some recently developed models on the MediQA-RQE test set based on the accuracy metric. The F1-score is also reported in parenthesis.

Ref.	Model	Accuracy
[62]	BioBERT + Named entity recognition	48.5
[57]	Ensemble approach (BERT + BioBERT + SVM)	48.9
[61]	Gensim Word2Vec + Siamese Network of Bidirectional LSTM	53.2
	ReQuEST[a]	52.2 (60.1)

[a] We supposed the γ coefficient equals 0, and then, examined ReQuEST performance with different combinations of $\alpha \in \{0.1, \cdots, 0.9, 1\}$ and $\beta \in \{0, 0.1, \cdots, 0.9\}$ coefficients. The reported result is the best one which was obtained with $\alpha = 0.5$ and $\beta = 0.5$, though other results were not much different.

TABLE 9. The performance comparison of some recently developed models on MeQSum test set in terms of Rouge-L (F1), Rouge-1 (F1), Rouge-2 (F1), and BERTScore (F1). Recall-based values are also reported in parentheses for our proposed method.

Ref.	Model	BERTScore	Rouge-L	Rouge-1	Rouge-2
[9]	BART	70.25	30.77	33.31	13.93
[63]	PEGASUS	69.96	31.49	33.40	15.99
[8]	Ensemble approach (PEGASUS + BART + T5) + Re-ranking + Error correction	68.98	31.31	35.14	16.08
	ReQuEST[a]	90.73 (90.79)	37.34 (44.39)	40.45 (48.34)	24.56 (29.69)

[a] Plenty of experiments were carried out using various coefficients, from which $\alpha = 0.3$, $\beta = 0.7$, and $\gamma = 0$ provided the best fit.

TABLE 10. The performance comparison of some recently developed models on Debatepedia test set. All Rouge values are recall-based except the ones in parenthesis, which are F1-based.

Ref.	Model	Rouge-L	Rouge-1	Rouge-2
[43]	Soft LSTM-based diversity attention	40.43	41.26	18.75
[64]	Sequence-to-sequence + LSTM + selection mechanism	42.73	43.22	27.40
[65]	BERTSUM + Query relevance	44.07	47.16	27.48
[66]	Sequence-to-sequence + Query relevance + pointer generator	46.18	53.09	16.10
	ReQuEST[a]	46.77 (47.30)	47.87 (48.45)	27.71 (28.00)

[a] In this experiment, the coefficients α, β, γ, ρ, and τ were randomly initialized with 0.3, 0.4, 0.3, 0.5, and 0.5, respectively.

proficient in all tasks. Nonetheless, the TG performance on test data is improved by about 11% and 2% in Rouge-L and BERTScore metrics, respectively. Furthermore, the test accuracy and F1-Score are both 1% enhanced. In addition, despite a slight degradation in the QFQS performance in terms of Rouge-L, the BERTScore criterion declares preserving the semantic quality of produced summaries. In conclusion, ReQuEST is expected to become more efficient by tuning more and reaching a compromise between tasks.

G. ANALYSIS OF REQUEST PERFORMANCE ON OTHER DATASETS

To answer RQ4 (*How efficient is ReQuEST on other datasets, whether the open domain or restricted domain?*), the performance of ReQuEST on three well-known datasets is compared with some recent approaches. MediQA-RQE is utilized for the RQE task, while MeQSum and Debatepedia are used for generic and query-focused abstractive summarization tasks. The test results are presented in Tables 8 to 10. It is worth noticing that aside from test data, each dataset includes training data on which ReQuEST parameters are trained. Moreover, those advanced approaches employing models pre-trained on in-domain data or incorporating specialized knowledge through data augmentation techniques are exempt from competition.

Table 8 summarizes the assessment results on the MediQA-RQE dataset. As Table 8 indicates, even pre-trained transformers on biological data, i.e., BioBERT, are beaten by the ReQuEST. However, the 1% supremacy of [61] suggests that the quality of text representation directly influences classifier performance.

Table 9 compares the performance of ReQuEST on the MeQSum dataset with several recent models. It signifies the superiority of ReQuEST over BART, PEGASUS, and their ensemble model in terms of BERTScore and Rouge metrics. In addition, gaining great BERTScore values indicates that summaries are well-generated, conveying the meaning of target sentences. Overall, the evidence confirms the effectiveness of multi-task learning for the QS task. In addition, it is hoped that incorporating domain-specific information or other question characteristics, such as question focus or question type, will improve the performance of ReQuEST.

Table 10 illustrates a performance comparison between ReQuEST and some competitors using the Debatepedia dataset. It is evident from this table that ReQuEST ranks the best in terms of Rouge-L. In addition, due to the simultaneous high values of Rouge-L and Rouge-1, it appears that ReQuEST can recover a considerable fraction of reference tokens in the same order. At the same time, Rouge-2 is less than a third, which means ReQuEST does not oblige itself to capture adjacent words. In contrast, it brings up to 90% BERTScore, which informs the high semantic quality of the created summaries. It is worthwhile to say that, though Debatepedia queries are sentences rather than sets of tags, ReQuEST handles them in the QFQS task. Besides, it generates them simultaneously through its TG task, resulting in approximately 55.7% Rouge-1, 38.6% Rouge-2, 53% Rouge-L, and 89.8% BERTScore metrics. In summary, the analyses signify that ReQuEST performs plausibly on different benchmarks. It is also anticipated that infusing more information will improve its performance.

V. CONCLUSION AND FUTURE WORK

In this paper, a transformer-based multi-task model called ReQuEST has been proposed to handle three essential tasks in community question-answering platforms, including question entailment recognition (RQE), question summarization (QS), and tag generation (TG). Leveraging BART as its backbone, ReQuEST predicts the entailment label of an input pair of questions and summarizes the first question in light of its tags. Meanwhile, it generates a sequence of tags for the first question, describing its main topics. Indeed, ReQuEST tends to minimize the total loss of all tasks while also making a compromise between them. Moreover, significantly reducing the number of parameters without depreciation of model performance is of great importance in the current research. Additionally, to the best of our knowledge, ReQuEST is the first effort that involves the TG alongside RQE and QFQS. As a consequence, more prosperous input representations are acquired by sharing the encoder component between all tasks. Besides, sharing the first layers of the decoder component between TG and QS allows them to benefit each other during reconstructing sequences. The experimental results on real-world datasets have indicated that ReQuEST can accurately predict the entailment relation and generate high-quality summaries and tag sequences.

In future work, we will address the following enhancements to the current study:

1) Assessing the effectiveness of other transformer models such as Longformer, T5, and DistillBART.
2) Identifying the best composition of coefficients by implementing an ML-based model.
3) Defining larger values for coefficients beyond the range of [0,1] to accelerate convergence.
4) Defining rouge-based objective functions instead of cross-entropy loss to better match the problem objectives.

REFERENCES

[1] B. Ojokoh and E. Adebisi, "A review of question answering systems," *J. Web Eng.*, vol. 17, no. 8, pp. 717–758, 2019, doi: 10.13052/jwe1540-9589.1785.

[2] S. Yadav, D. Gupta, A. Ben Abacha, and D. Demner-Fushman, "Reinforcement learning for abstractive question summarization with question-aware semantic rewards," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 249–255, doi: 10.18653/v1/2021.acl-short.33.

[3] K. Mrini, F. Dernoncourt, S. Yoon, T. Bui, W. Chang, E. Farcas, and N. Nakashole, "A gradually soft multi-task and data-augmented approach to medical question understanding," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 1505–1515, doi: 10.18653/v1/2021.acl-long.119.

[4] D. Hoogeveen, L. Wang, T. Baldwin, and K. M. Verspoor, "Web forum retrieval and text analytics: A survey," *Found. Trends Inf. Retr.*, vol. 12, no. 1, pp. 1–163, 2018, doi: 10.1561/1500000062.

[5] K. Balog, A. Schuth, P. Dekker, and N. Tavakolpoursaleh, "Overview of the TREC 2016 open search track," in *Proc. 24th Text Retr. Conf.*, 2016, pp. 1–9.

[6] C. M. Suneera and J. Prakash, "A BERT-based question representation for improved question retrieval in community question answering systems," in *Advances in Machine Learning and Computational Intelligence: Algorithms for Intelligent Systems*. Singapore: Springer, 2021, pp. 341–348, doi: 10.1007/978-981-15-5243-4_31.

[7] S. Yadav, D. Gupta, A. B. Abacha, and D. Demner-Fushman, "Question-aware transformer models for consumer health question summarization," *J. Biomed. Informat.*, vol. 128, Apr. 2022, Art. no. 104040, doi: 10.1016/j.jbi.2022.104040.

[8] Y. He, M. Chen, and S. Huang, "Damo_nlp at MEDIQA 2021: Knowledge-based preprocessing and coverage-oriented reranking for medical question summarization," in *Proc. 20th Workshop Biomed. Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 112–118, doi: 10.18653/v1/2021.bionlp-1.12.

[9] S. Balumuri, S. Bachina, and S. Kamath, "SB_NITK at MEDIQA 2021: Leveraging transfer learning for question summarization in medical domain," in *Proc. 20th Workshop Biomed. Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 273–279, doi: 10.18653/v1/2021.bionlp-1.31.

[10] P. K. Roy and J. P. Singh, "A Tag2Vec approach for questions tag suggestion on community question answering sites," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, 2018, pp. 168–182, doi: 10.1007/978-3-319-96133-0_13.

[11] I. Saleh and N. El-Tazi, "Automatic organization of semantically related tags using topic modelling," in *Proc. Eur. Conf. Adv. Databases Inf. Syst.*, 2017, pp. 235–245, doi: 10.1007/978-3-319-67162-8_23.

[12] B. Yang and S. Manandhar, "Tag-based expert recommendation in community question answering," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 960–963, doi: 10.1109/ASONAM.2014.6921702.

[13] A. Ben Abacha and D. Demner-Fushman, "A question-entailment approach to question answering," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–23, Dec. 2019, doi: 10.1186/s12859-019-3119-4.

[14] B. Li, P. Yang, H. Zhao, P. Zhang, and Z. Liu, "Hierarchical sliding inference generator for question-driven abstractive answer summarization," *ACM Trans. Inf. Syst.*, vol. 41, no. 1, pp. 1–27, Jan. 2023, doi: 10.1145/3511891.

[15] D. Hoogeveen, A. Bennett, Y. Li, K. Verspoor, and T. Baldwin, "Detecting misflagged duplicate questions in community question-answering archives," in *Proc. 12th Int. AAAI Conf. Web Soc. Media*, Jun. 2018, vol. 12, no. 1, pp. 112–120, doi: 10.1609/icwsm.v12i1.15011.

[16] B. Xu, Z. Xing, X. Xia, and D. Lo, "AnswerBot: Automated generation of answer summary to developers' technical questions," in *Proc. 32nd IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Oct. 2017, pp. 706–716, doi: 10.1109/ASE.2017.8115681.

[17] R. Zhang, Q. Zhou, B. Wu, W. Li, and T. Mo, "What do questions exactly ask? MFAE: Duplicate question identification with multi-fusion asking emphasis," in *Proc. SIAM Int. Conf. Data Mining*, 2020, 2020, pp. 226–234, doi: 10.1137/1.9781611976236.26.

[18] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft, "Asking clarifying questions in open-domain information-seeking conversations," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 475–484, doi: 10.1145/3331184.3331265.

[19] R. Cai, B. Zhu, L. Ji, T. Hao, J. Yan, and W. Liu, "An CNN-LSTM attention approach to understanding user query intent from online health communities," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 430–437, doi: 10.1109/ICDMW.2017.62.

[20] X. Shi, H. Huang, S. Zhao, P. Jian, and Y.-K. Tang, "Tag recommendation by word-level tag sequence modeling," in *Proc. Int. Conf. Database Syst. Advanced Appl.*, Nov. 2019, pp. 420–424, doi: 10.1007/978-3-030-18590-9_58.

[21] X. Zhang, M. Liu, J. Yin, Z. Ren, and L. Nie, "Question tagging via graph-guided ranking," *ACM Trans. Inf. Syst.*, vol. 40, no. 1, pp. 1–23, Jan. 2022, doi: 10.1145/3468270.

[22] L. Nie, Y. Li, F. Feng, X. Song, M. Wang, and Y. Wang, "Large-scale question tagging via joint question-topic embedding learning," *ACM Trans. Inf. Syst.*, vol. 38, no. 2, pp. 1–23, Apr. 2020, doi: 10.1145/3380954.

[23] H. Guo, R. Pasunuru, and M. Bansal, "Soft layer-specific multi-task summarization with entailment and question generation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 687–697, doi: 10.18653/v1/p18-1064.

[24] Y. Wu, W. Wu, Z. Li, and M. Zhou, "Improving recommendation of tail tags for questions in community question answering," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2016, vol. 30, no. 1, pp. 1–7, doi: 10.1609/aaai.v30i1.10367.

[25] D. Ganguly and G. J. F. Jones, "Partially labeled supervised topic models for retrieving similar questions in CQA forums," in *Proc. Int. Conf. Theory Inf. Retr.*, New York, NY, USA, Sep. 2015, pp. 161–170, doi: 10.1145/2808194.2809460.

[26] R. C. Belwal, S. Rai, and A. Gupta, "Text summarization using topic-based vector space model and semantic measure," *Inf. Process. Manag.*, vol. 58, no. 3, May 2021, Art. no. 102536, doi: 10.1016/j.ipm.2021.102536.

[27] S. Liu, J. Cao, R. Yang, and Z. Wen, "Key phrase aware transformer for abstractive summarization," *Inf. Process. Manag.*, vol. 59, no. 3, May 2022, Art. no. 102913, doi: 10.1016/j.ipm.2022.102913.

[28] T. Falke, L. F. R. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych, "Ranking generated summaries by correctness: An interesting but challenging application for natural language inference," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 2214–2220, doi: 10.18653/v1/p19-1213.

[29] H. Li, J. Zhu, J. Zhang, and C. Zong, "Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1430–1441.

[30] R. Pasunuru and M. Bansal, "Multi-reward reinforced summarization with saliency and entailment," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 646–653, doi: 10.18653/v1/n18-2102.

[31] J. Liu and Y. Yang, "Enhancing summarization with text classification via topic consistency," in *Machine Learning and Knowledge Discovery in Databases. Research Track*, vol. 12977. Cham, Switzerland: Springer, 2021, pp. 661–676, doi: 10.1007/978-3-030-86523-8_40.

[32] A. Ben Abacha and D. Demner-Fushman, "On the summarization of consumer health questions," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 2228–2234, doi: 10.18653/v1/p19-1215.

[33] M. S. Zahedi, M. Rahgozar, and R. A. Zoroofi, "HCA: Hierarchical compare aggregate model for question retrieval in community question answering," *Inf. Process. Manag.*, vol. 57, no. 6, Nov. 2020, Art. no. 102318, doi: 10.1016/j.ipm.2020.102318.

[34] V. B. Kumar, A. Srinivasan, A. Chaudhary, J. Route, T. Mitamura, and E. Nyberg, "Dr.Quad at MEDIQA 2019: Towards textual inference and question entailment using contextualized representations," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2019, pp. 453–461, doi: 10.18653/v1/W19-5048.

[35] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Jan. 2022, doi: 10.1109/TKDE.2020.2981314.

[36] H. Pugaliya, K. Saxena, S. Garg, S. Shalini, P. Gupta, E. Nyberg, and T. Mitamura, "Pentagon at MEDIQA 2019: Multi-task learning for filtering and re-ranking answers using language inference and question entailment," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 389–398, doi: 10.18653/v1/w19-5041.

[37] L. Li, M. Zhang, Z. Chao, and J. Xiang, "Using context information to enhance simple question answering," *World Wide Web*, vol. 24, no. 1, pp. 249–277, Jan. 2021, doi: 10.1007/s11280-020-00842-7.

[38] M. Sarrouti, A. B. Abacha, and D. Demner-Fushman, "Multi-task transfer learning with data augmentation for recognizing question entailment in the medical domain," in *Proc. IEEE 9th Int. Conf. Healthcare Informat. (ICHI)*, Aug. 2021, pp. 339–346, doi: 10.1109/ICHI52183.2021.00058.

[39] H. Zhou, X. Li, W. Yao, C. Lang, and S. Ning, "DUT-NLP at MEDIQA 2019: An adversarial multi-task network to jointly model recognizing question entailment and question answering," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 437–445, doi: 10.18653/v1/w19-5046.

[40] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[41] D. Hoogeveen, K. M. Verspoor, and T. Baldwin, "CQADupStack: A benchmark data set for community question-answering research," in *Proc. 20th Australas. Document Comput. Symp.*, New York, NY, USA, Dec. 2015, pp. 1–8, doi: 10.1145/2838931.2838934.

[42] A. B. Abacha, C. Shivade, and D. Demner-Fushman, "Overview of the MEDIQA 2019 shared task on textual inference, question entailment and question answering," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 370–379, doi: 10.18653/v1/W19-5039.

[43] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1063–1072, doi: 10.18653/v1/p17-1098.

[44] D. Jani, N. Patel, H. Yadav, S. Suthar, and S. Patel, "A concise review on automatic text summarization," in *Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies*, vol. 281. Singapore: Springer, 2022, pp. 523–536, doi: 10.1007/978-981-16-9447-9_40.

[45] Y. Deng, W. Lam, Y. Xie, D. Chen, Y. Li, M. Yang, and Y. Shen, "Joint learning of answer selection and answer summary generation in community question answering," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 5, pp. 7651–7658, doi: 10.1609/aaai.v34i05.6266.

[46] H. Van Lierde and T. W. S. Chow, "Query-oriented text summarization based on hypergraph transversals," *Inf. Process. Manag.*, vol. 56, no. 4, pp. 1317–1338, Jul. 2019, doi: 10.1016/j.ipm.2019.03.003.

[47] D. Su, T. Yu, and P. Fung, "Improve query focused abstractive summarization by incorporating answer relevance," in *Proc. Findings Assoc. Comput. Linguistics, ACL-IJCNLP*. Stroudsburg, PA, USA: Association for Computational Linguistics, May 2021, pp. 3124–3131, doi: 10.18653/v1/2021.findings-acl.275.

[48] M. T. R. Laskar, E. Hoque, and J. Huang, "Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models," in *Advances in Artificial Intelligence*, vol. 12109. Cham, Switzerland: Springer, 2020, pp. 342–348, doi: 10.1007/978-3-030-47358-7_35.

[49] Y. Xu and M. Lapata, "Document summarization with latent queries," *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 623–638, May 2022, doi: 10.1162/tacl_a_00480.

[50] K. Lei, Q. Fu, M. Yang, and Y. Liang, "Tag recommendation by text classification with attention-based capsule network," *Neurocomputing*, vol. 391, pp. 65–73, May 2020, doi: 10.1016/j.neucom.2020.01.091.

[51] Y. Wu, S. Xi, Y. Yao, F. Xu, H. Tong, and J. Lu, "Guiding supervised topic modeling for content based tag recommendation," *Neurocomputing*, vol. 314, pp. 479–489, Nov. 2018, doi: 10.1016/j.neucom.2018.07.011.

[52] J. He, B. Xu, Z. Yang, D. Han, C. Yang, and D. Lo, "PTM4Tag: Sharpening tag recommendation of stack overflow posts with pre-trained models," in *Proc. 30th IEEE/ACM Int. Conf. Program Comprehension*, New York, NY, USA, May 2022, pp. 1–11, doi: 10.1145/3524610.3527897.

[53] K. K. Pal, M. Gamon, N. Chandrasekaran, and S. Cucerzan, "Modeling tag prediction based on question tagging behavior analysis of CommunityQA platform users," Jul. 2023, *arXiv:2307.01420*.

[54] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.

[55] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, May 2020, vol. 33, pp. 1877–1901.

[56] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[57] V. Nguyen, S. Karimi, and Z. Xing, "ANU-CSIRO at MEDIQA 2019: Question answering using deep contextual knowledge," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 478–487, doi: 10.18653/v1/w19-5051.

[58] W. Zhu, X. Zhou, K. Wang, X. Luo, X. Li, Y. Ni, and G. Xie, "PANLP at MEDIQA 2019: Pre-trained language models, transfer learning and knowledge distillation," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 380–388, doi: 10.18653/v1/w19-5040.

[59] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Work. Text Summ. Branches Out (WAS)*, no. 1, 2004, pp. 25–26.

[60] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," Apr. 2019, *arXiv:1904.09675*.

[61] D. Bandyopadhyay, B. Gain, T. Saikh, and A. Ekbal, "IITP at MEDIQA 2019: Systems report for natural language inference, question entailment and question answering," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 517–522, doi: 10.18653/v1/w19-5056.

[62] A. Lamurias and F. M. Couto, "LasigeBioTM at MEDIQA 2019: Biomedical question answering using bidirectional transformers and named entity recognition," in *Proc. 18th BioNLP Workshop Shared Task*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 523–527, doi: 10.18653/v1/w19-5057.

[63] M. Sänger, L. Weber, and U. Leser, "WBI at MEDIQA 2021: Summarizing consumer health questions with generative transformers," in *Proc. 20th Workshop Biomed. Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 86–95, doi: 10.18653/v1/2021.bionlp-1.9.

[64] C. Aryal and Y. Chali, "Selection driven query focused abstractive document summarization," in *Proc. 33rd Canadian Conf. Artif. Intell.*, 2020, pp. 118–124, doi: 10.1007/978-3-030-47358-7_11.

[65] D. M. Abdullah and Y. Chali, "Towards generating query to perform query focused abstractive summarization using pre-trained model," in *Proc. 13th Int. Conf. Natural Lang. Gener.*, 2020, pp. 80–85.

[66] T. Baumel, M. Eyal, and M. Elhadad, "Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models," Jan. 2018, *arXiv:1801.07704*.

SEYYEDE ZAHRA AFTABI received the B.S. degree in software engineering from Shahid Rajaee University, Tehran, Iran, in 2013, and the M.S. degree in software engineering from Yazd University, Yazd, Iran, in 2016. She is currently pursuing the Ph.D. degree in artificial intelligence with the Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran. Her research interests include information retrieval, text mining, question-answering systems, and natural language processing using deep learning models.

SEYYEDE MARYAM SEYYEDI received the B.S. degree in computer engineering from the K. N. Toosi University of Technology, Tehran, Iran, in 2020, where she is currently pursuing the M.Sc. degree with the Faculty of Computer Engineering. Her research interests include natural language processing, single- and multi-document summarization, text entailment, and creating synthetic datasets.

MOHAMMAD MALEKI was born in Tehran, Iran, in 1997. He received the B.S. degree in computer engineering from the K. N. Toosi University of Technology, Tehran, in 2020, where he is currently pursuing the M.Sc. degree in software engineering.

His career contains five years of full-stack web development. His current research interests include automatic text summarization and textual entailment.

SAEED FARZI received the Ph.D. degree in computer engineering from Tehran University, Tehran, Iran, in 2016.

He joined the Artificial Intelligence Department, K. N. Toosi University of Technology, Tehran, in 2017. His research interests include machine learning, information retrieval, and social network analysis.