Received ; | Revised ; | Accepted

DOI: xxx/xxxx

ARTICLE TYPE

Question-Driven Text Summarization Using an Extractive-Abstractive Framework

Mahsa Abazari Kia*[1] | Aygul Garifullina[2] | Mathias Kern[2] | Jon Chamberlain[1] | Shoaib Jameel[3]

[1]School of Computer Science and Electronic Engineering, University of Essex, UK
[2]BT, Adastral Park, Ipswich, UK
[3]Department of Electronics and Computer Science, University of Southampton, UK

Correspondence
*Email: ma19194@essex.ac.uk

Abstract

Question-driven automatic text summarization is a popular technique to produce concise and informative answers to specific questions using a document collection. Both query-based and question-driven summarization may not produce reliable summaries nor contain relevant information if they do not take advantage of extractive and abstractive summarization mechanisms to improve performance. In this paper, we propose a novel extractive and abstractive hybrid framework designed for question-driven automatic text summarization. The framework consists of complimentary modules that work together to generate an effective summary: (1) discovering appropriate non-redundant sentences as plausible answers using an open-domain multi-hop question answering system based on a Convolutional Neural Network (CNN), multi-head attention mechanism and reasoning process; and (2) a novel paraphrasing Generative Adversarial Network (GAN) model based on transformers rewrites the extracted sentences in an abstractive setup. Experiments show this framework results in more reliable abstractive summary than competing methods. We have performed extensive experiments on public datasets, and the results show our model can outperform many question-driven and query-based baseline methods (an R1, R2, RL increase of 6%-7% for over the next highest baseline).

KEYWORDS:
Hybrid text summarization, Multi-hop QA, Question-driven text summarization, Abstractive text summarization

1 | INTRODUCTION

With the advent of the Internet and social media, we have seen an exponential rise in text generated every day. It is difficult to keep pace with the quantity of information conveyed. By reducing extraneous material from documents, text summarizing minimizes the time to comprehend the content, allowing a reader to focus on important aspects rapidly. Shortening a text document while maintaining its overall meaning and information conveyed is the goal of text summarization [1]. Automatic Text Summarization (ATS) systems are categorized as single-document or multi-document systems [2]. The former generates a single document summary, whereas the latter generates a summary for a group of documents. ATS systems are created by employing either an extractive or abstractive approach. Another class of models jointly models abstractive and extractive paradigms, mainly hybrid approaches. In extractive models, the goal is to extract sentences or pieces of text from the document to optimize the information conveyed by removing redundancy. The task of rephrasing the language by comprehending the semantic information conveyed by the documents is known as abstractive text summarization, which necessitates a thorough understanding of natural language

TABLE 1 Query-based and question-driven text summarization example

Text: ... **According to Powers, your eating plan should focus on the type and amount of carbohydrates you eat throughout the day. Choose low-carb vegetables such as mushrooms, onions, eggplant, tomatoes, Brussels sprouts, and zucchini, as well as low-carb squashes. To add flavor and texture to a meal, serve them with low-fat souces, hummus, guacamole, and salsa, or roasted with herbs and spices like rosemary, cayenne pepper, and garlic. Sweet potatoes, when combined with other meals, can successfully slow food digestion, increase satiety, and moderate blood sugar swings. Thus, individuals who are hyperglycemic can have some sweet potatoes, which will not only not elevate blood sugar but will also aid in blood sugar control.** ...
Query: foods for lower blood sugar
Query-based Gold Summary: The amount and type of carbs you put in your diet throughout the day should be seriously considered. Low-carb and tasty veggies, like mushrooms, onions, eggplant, tomatoes, Brussels sprouts, and low-carb squashes, like zucchini with dips such as low-fat dressings, hummus, guacamole, and salsa, or roasted with different seasonings such as rosemary, cayenne pepper, or garlic could be included to the meal for better flavor and texture. Sweet potatoes can help to slow down food digestion, increase satiety, and stabilise blood sugar levels which not only do not raise blood sugar but also help to control blood sugar.
Question: How sweet potatoes helps people with hyperglycemic?
Question-driven Gold Summary: Sweet potatoes can help to slow down food digestion, increase satiety, and stabilise blood sugar levels. As a result, persons with hyperglycemia can eat sweet potatoes, which not only do not raise blood sugar but also help to control blood sugar.

processing[3]. Abstractive text summarization can be regarded as more appealing than extractive summarization, but it is challenging to perform because it requires the capability of generating new sentences[4]. Hybrid approaches combine extractive, and abstractive models, which exploit the complementary advantages between the two[5].

Compared to extractive summarization, the content produced by abstractive methods often suffers issues such as data redundancy, poor readability, and major semantic diversion from the source document(s). The majority of contemporary abstractive summarization models are built on neural networks with sequence-to-sequence (seq2seq)[6,7,8,9,10,11]. They are composed of encoders for the purpose of comprehending the input sequence and decoders for the purpose of generating the output sequence. However, there are four significant drawbacks to generating reasonable text with seq2seq neural networks: (1) the out-of-vocabulary (OOV) problem; (2) continually producing a particular word or phrase, which introduces redundancy; (3) test-time exposure bias; and (4) non-optimized learning for evaluation metrics used by models in disciplines such as text summarization and machine translation. As a result, they cannot generate appropriate abstractive summaries since they cannot convey the semantics of the document[12,13]. To recreate key content, abstractive summarization requires advanced natural language techniques for reading and understanding the text. In contrast, the extractive summaries may include repeated terms, high frequency of particular phrases, and redundancy in some sentences[2].

Text summarizers have applications in different domains: generic (domain-independent); specific (domain-dependent); opinion/sentiment summarization; query-based summarization; and question-based summarization[14]. Generic (domain-independent) text summarization provides a brief overview of a long document, conveying the core message of the document[2] and summarizes documents that are from different domains[15]. The domain-specific ATS systems, on the other hand, are designed to summarize documents within a specific domain (e.g., legal documents[14,16], or medical reports[17,18,19]). Opinion summarization refers to the process of automatically summarizing multiple opinions that discuss the same subject[20]. A query-based summarization approach summarizes query-related content from the source document(s)[21]. Question-driven summarization approaches answer a question and also provide additional informative content to that answer from the source document(s) to make it more understandable and convincing[22]. A question-driven summary must satisfy three goals: answerability, understandability, and persuasiveness. There have been several attempts to develop methods for question-driven automatic text summarization[22,23,24,25,26]. Examples of query-based and question-based text summarization are provided in Table 1. The bold text shows the relevant text to the proposed query and question, and the gold summary is the abstractive summary generated by a human. The query-based summary has summarized the text given the query "foods for lower blood sugar". It contains all the information about the diet for hyperglycemic people. Still, the question-driven summary is shorter and only contains specific information, the answer and its explanation, to the question.

Query-based summarization techniques use semantic relevance measurement to summarize the query-related content from the source document [27,28,29,30]. These approaches are not suitable for tackling question-driven summarization problems in Question Answering (QA) scenarios. For question-driven summarization, answer detection and the reasoning on the detected answer are needed. To overcome the problems mentioned above and acquire a reliable summary of the document regarding a question, we propose a two-stage, hybrid extractive and abstractive summarization that combines the advantages of the two methods. Firstly,

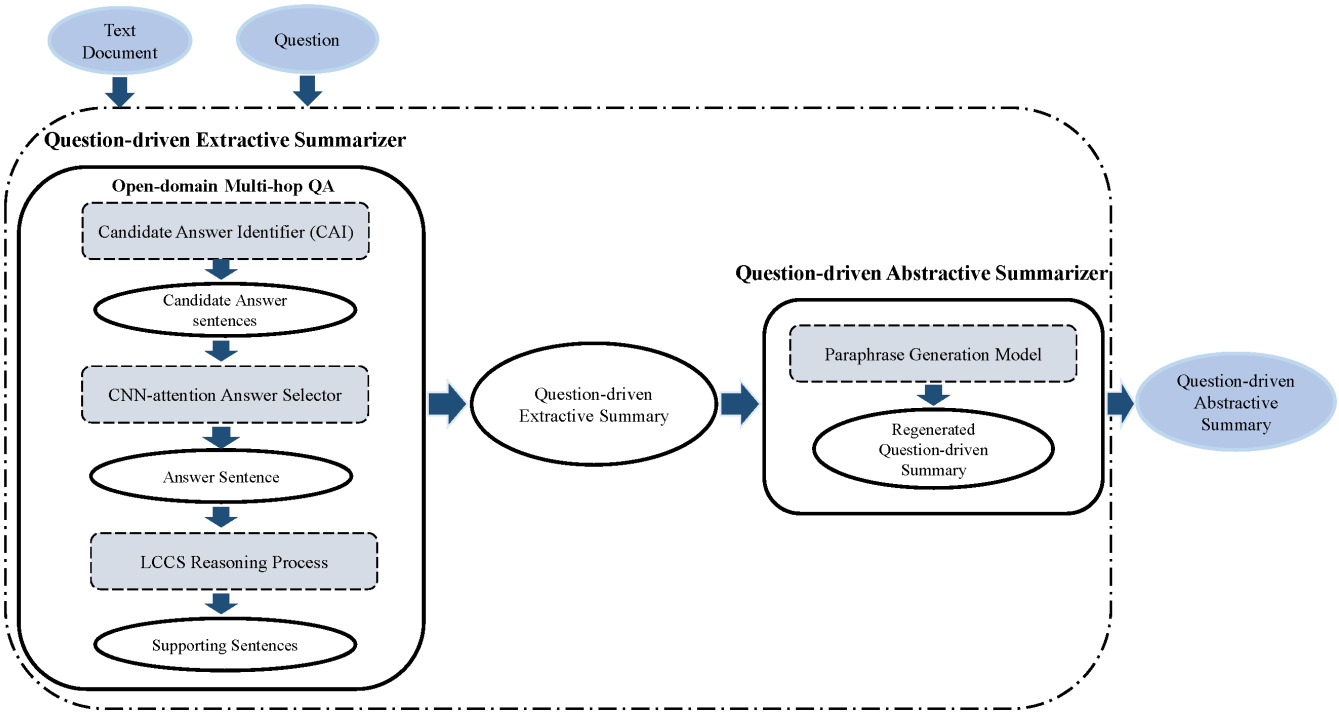

FIGURE 1 Our proposed hybrid question-driven text summarization framework.

the extractive model selects the answer sentence and its supporting sentences, which provide details or explanations for the answer sentence. After obtaining the question-driven extractive summary, a novel abstractive model transforms the extractive summary into an abstractive summary. The extractive phase reduces the amount of redundant information, which improves the effectiveness of the abstractive summarizer. We propose a question-driven abstractive summarization model, depicted in Fig. 1 which we describe in detail later, and the main contributions can be summarized as follows:

1. A novel open-domain multi-hop QA model based on a CNN and multi-head attention mechanism designed to comprehend the document and question for constructing the question-driven extractive summary. The answer selector module measures the semantic dependencies between the local features extracted from the document sentences and question to select the answer sentence.

2. A novel reasoning approach is proposed for analyzing the document regarding the detected answer sentence and searching for relevant supporting sentences based on lexical coverage and contextual semantic similarity.

3. A novel paraphrase framework based on General Adversarial Networks (GANs), Q-learning, and transformers to produce question-driven abstractive summaries from the generated extractive summaries. We showed that rewriting the extractive summaries using a paraphrase generation model helps us have abstractive summaries closer to gold summaries (human-generated summaries).

2 | RELATED WORK

In this section, we cover closely related literature. We also mention how our proposed method is different from the existing methods. Since our approach is a question-driven hybrid text summarization based on GANs for generating abstractive summaries, we have covered the existing GAN methods for text generation, query-based summarization, hybrid text summarization, abstractive summarization, and paraphrase generation approaches in the following subsections.

2.1 | Generative Adversarial Networks for Text Generation

GAN was first applied in the computer vision domain. Applying GAN to text generation is nontrivial because GAN is designed for generating real-valued, continuous data but struggles to generate discrete token sequences, such as texts. However, several studies are proposed to tailor this powerful network for text generation; therefore, we have summarized recent approaches and their advantages and drawbacks in this section. Generative Adversarial Nets (GAN) was proposed in[31]. It consists of two simultaneously trained models: one (the generator) trained to create new data, and the other (the discriminator) trained to distinguish the generated data (fake data) from real examples. Yu et al.,[32] introduced the first Reinforcement Learning (RL)-based[33] work, called Sequence Generative Adversarial Network (SeqGAN) which Monte Carlo tree search (MCTS) is used to calculate the reward at every generation step for evaluating a generated subsequence which is computationally expensive but in StepGAN proposed by Tuan et al.,[34], the discriminator is altered to automatically provide scores at each generation stage for assessing the quality of each subsequence. In StepGAN, a seq2seq generator and discriminator are designed, and the discriminator predicts the immediate rewards using Q-learning[35] without performing a tree search. TextGAIL[36] improves the discriminator's guidance by combining RoBERTa and GPT-2[37] with recent advances in RL. Zhang et al.,[38], enhance the standard actor-critic methodology[39] by designing a transformer-based generator and a CNN-based discriminator. A key barrier is a language's inherent characteristics, such as syntax, grammar, and semantic aspects. The model must learn the correct connection between words and characters to generate a viable text, commonly accomplished through various memories and situations (prior knowledge). Such issues can be addressed in a more robust pre-learning step, in which pre-trained embedding models BERT[40], A lite bert for self-supervised learning of language representations (ALBERT)[41], ELECTRA[42], or GPT-2 are combined with transformer-based seq2seq architectures to be capable of generating plausible "natural" language text. Transformer-based GANs incorporating contextualized pre-trained language models and stepwise evaluation are blank spots that still need to be appropriately addressed for text generation, which we have presented in this paper.

2.2 | Query-based Text Summarization

As we present a deep-learning-based approach in this paper, we describe recent approaches based on neural networks. Nema et al.,[29] introduced a typical encode-attend-decode model (based on LSTM) for query-based abstractive summarization, which first computes a vectorial representation for the document and the query, and then the decoder produces a contextual summary one word at a time. Li et al.,[43] designed a bi-GRU sentence-level encoder is proposed to encode a sentence in a document, and then a query filter component attention model upon the sentence encoder is designed to inject such information into sentence encoding and computing the new sentence encoding, including the query information. In the end, a feed-forward neural network is applied to compute a salience score for each sentence. Ishigaki et al.,[30] introduced three copying mechanisms designed for query-based abstractive summarizers. In the copying mechanism, two different probabilities for every word in the vocabulary are considered, the copying probability and the generation probability. Zhao et al.,[44] have designed 3 solutions for Chinese query-based document summarization utilizing relevance ranking, dual attention and pre-trained word embeddings, BERT-based encoder and a text-pair classification which performed better than the other two methods. We have studied recent query-based text summarization approaches in this section to examine their ability to be used for question-driven text summarization. The main goal in query-based text summarization approaches is to summarize the retrieved relevant information to the query, but in the question-driven text summarization, answer detection and explaining that answer in a summarized form is desired. Furthermore, the query-based text summarization approaches are not adaptable for the question-driven summarization problem.

2.3 | Abstractive Text Summarization

Various approaches are proposed for abstractive summarization, but here we only consider those based on GANs, which are more relevant to our work. Liu et al.,[45] used RL (i.e., policy gradient) to optimize the bi-directional LSTM generator and implemented the discriminator as a trained text classifier to classify the generated summaries as a machine or human-generated. Scialom et al.,[46] introduced an approach using Discriminative Adversarial Search (DAS) utilizing the Unified Language Model for natural language understanding and generation (UniLM)[47] based on BERT for generator. The seq2seq based discriminator is integrated into a beam search that obtains a label at each generation step to refine the probabilities and select the top candidate sequences. Rekabdar et al.,[48] designed a generator based on LSTM encoder-decoder with an attention mechanism which is modeled as a stochastic policy in RL, and the discriminator is based on CNN. Dang et al.,[49] presented a GAN model containing one generator and two discriminators. The generator is based on LSTM encoder-decoder and one of the discriminators

is the similarity discriminator based on CNN text classifier with four classes (Incomplete class, Redundant class, Similar class, Irrelevant class). The other one is Readability Discriminator, a CNN-based model which tells whether the generator or human generates the summary.

Our method is significantly different from these methods in several ways, and it consists of two main components, extractive summarizer, and abstractive summarizer. The extractive component filters the irrelevant information and feeds the pruned information (extractive summary) to the abstractive component. We have designed a transformer-based GAN with Q-stepwise evaluation for abstractive part which regenerates and rewrites the generated extractive summaries and produces reliable abstractive summaries. Using transformers architectures with GAN and applying stepwise evaluation for generating text is an unexplored architecture which we proposed and studied in this paper.

2.4 | Hybrid Text Summarization

The hybrid approaches combine the abstractive and extractive approaches and their advantages[2]. Wang et al.,[50] proposed a hybrid system "EA-LTS" comprises two stages, the extractive stage selects the key sentences using a graph model, and the abstractive stage is an RNN based encoder-decoder in addition to an attention and pointer mechanisms to generate summaries. Bhat et al.,[51] proposed "SumItUp" for single-document summarization consisting of an extractive sentence selection based on statistical features and an abstractive summary generation for converting extractive summary to abstractive using a language generator. Subramanian et al.,[52] created a basic extraction step utilizing a hierarchical bidirectional LSTM seq2seq sentence pointer. This phase minimizes the amount of context for a following abstractive step with a single trained transformer language model. Chen et al.,[53] presented a framework composed of five components: (1) Word-level bidirectional GRU encoder for encoding the sentences word-by-word, (2) Sentence-level bidirectional GRU encoder encodes the document sentences, (3) Sentence extractor for labeling each sentence, (4) Hierarchical attention facilitates generating the sentence-level and word-level context vectors to be consumed in the decoding steps, (5) A GRU-based decoder with a beam search algorithm decodes the output word sequence. Jin et al.,[54] unified extractive and abstractive summarization into one architecture based on attention mechanism. Extractive summarization works on sentence granularity and directly conducts the sentence representations, while abstractive summarization is designed for operating on word granularity and their representations. Our work is different from the above approaches in several ways. First, we generate an extractive summary using the proposed Multi-hop QA system and relevance ranking method, then a paraphrase generation model is designed for transforming the extractive summary to abstractive.

2.5 | Paraphrase Generation

The task of paraphrase generation refers to generating one or multiple candidate paraphrases given the input sentence, which requires that the generated sentence and input sentence are different in expression form, but have the same expressed meaning[55]. Li et al.,[56] introduced DNPG as a way to decompose a sentence into sentence-level and phrase-level patterns in order to make neural paraphrase creation more intelligible and controlled, and they observed that DNPG could be applied to unsupervised domain adaptation for paraphrase production. Fu et al.,[57] suggested a novel model of paraphrasing based on a latent bag of words. Siddique et al.,[58] suggested an unsupervised paraphrase model using a variational autoencoder in a deep reinforcement learning framework. Liu et al.,[59] regarded paraphrase generation as an optimization issue and created a complex objective function. All of the strategies outlined above are concerned with the general quality of paraphrases and are unconcerned with their variety. Yang et al.,[60],Cao et al.,[61], Vizcarra et al.,[62],Tuan et al.,[34] proposed paraphrase generation models based on GAN which we consider them as our baseline methods and discussed them in detail in section 4.2. Paraphrase generation is a fundamental task of natural language processing (NLP) that has been broadly used in many downstream applications, such as information retrieval, machine translation, question answering and so on. This is the first work that employs a paraphrase generation model for generating abstractive summaries to the best of our knowledge. To this end, we have proposed a novel paraphrase model based on transformers and Q-learning stepwise evaluation for text generation, which is an unexplored architecture.

3 | OUR NOVEL QUESTION-DRIVEN HYBRID TEXT SUMMARIZATION MODEL

We proposed a hybrid text summarization approach for generating question-driven extractive-abstractive summaries. Our inputs are a text document and a question as depicted in Fig. 1. We developed an open-domain multi-hop QA system to select the

answer sentence and extract the supporting sentences for generating the question-driven extractive summary. To generate high-quality summaries for human consumption, a novel paraphrase generation model is proposed to rewrite the sentences of the extractive summary and construct a question-driven abstractive summary. In a nutshell, in our novel framework, we exploit the advantages of both extractive and abstractive models. Our novel extractive model automatically selects the most appropriate sentences from the document that conveys non-redundant and important information to the question. Subsequently, our novel abstractive paraphrasing model uses GAN and transformers to generate high-quality abstractive summaries so that the resulting summaries are coherent and readable. As mentioned above, a key advantage of our model is that the extractive phase helps remove redundant information which not only helps improve the quality of the summary generated by our abstractive summarizer but also makes it efficient because the abstractive phase does not have to deal with a large amount of data.

In the subsection below, we describe our question-driven extractive summary model followed by the question-driven abstractive model.

3.1 | Question-driven Extractive Model

We have proposed a question-driven extractive summarizer based on an open-domain multi-hop QA system comprising candidate answer identifier (CAI), answer sentence selector, and reasoning process. We have utilized CAI module introduced in[63] which has six functions based on linguistic and syntactic features and patterns for reducing the document to sentences (candidate answer sentences) that could answer the given question. We designed a joint CNN and multi-head attention neural network to analyze and assign a score to each candidate answer sentence based on its relevance to the question. The CNN-attention layer calculates the relevance score based on the correlation of the semantic features extracted from the question and the candidate answer sentence.

After selecting the answer sentence, an unsupervised reasoning process (we call it *LCCS* reasoning process) based on ***L***exical ***C***overage and ***C***ontextualized ***S***imilarity for selecting supporting sentences (justification sentences). The reasoning process helps us select appropriate, relevant sentences explaining the answer sentence and then constructs the final extractive summary. The overall framework of our extractive model is shown in Fig. 2. We rearrange the justification sentences and answer sentence according to their original indexes in the given document to bring coherence in the selected sequence of sentences and generate the question-driven extractive summary.

3.1.1 | Candidate Answers Identifier (CAI)

The candidate Answer Identifier (CAI) module limits the document to the sentences that are capable to answer the question based on its category (*When*, *Where*, *Who*, *What*, *Why*, *How*). CAI contains six functions for classifying the document sentences based on their linguistic and syntactic features. Preprocessing and splitting the document into sentences is the first step in the CAI module. Then each sentence will be analyzed to be assigned to one or more question categories based on their linguistic features, shown in Fig. 3. NN, NNP, NNS, NNPS, PRP, VB, VBG, and IN stand for a singular noun, singular proper noun, plural noun, plural proper noun, personal pronoun, verb, present participle verb, and preposition or subordinating conjunction respectively.

3.1.2 | CNN Multi-head Attention-based Answer Selector

There are K possible candidate answers $CA_1, CA_2, \cdots, CA_k$ in the document D for the given question q. The question q and candidate answer CA_i are concatenated into a single sequence with m and n tokens. The special token $[SEP]$ is used to denote the distinction between the question and candidate answer in the CNN multi-head attention layer's input. For the embedding layer, we get the semantic representation of q and CA_i using a pre-trained ALBERT contextual language model. The objective is to obtain the most trustworthy answer sentence CA_j to the q in D. The output of ALBERT is taken only for the first token $[CLS]$ which is used as the aggregate representation of the sequence. Untrained layers of CNN, pooling, and multi-head attention are added for fine-tuning the pre-trained ALBERT model. The CNN and attention mechanism constructs the question and sentence representation by focusing on the most significant features and their connections. The model produces a score for the candidate answer sentence utilizing the constructed semantic representation.

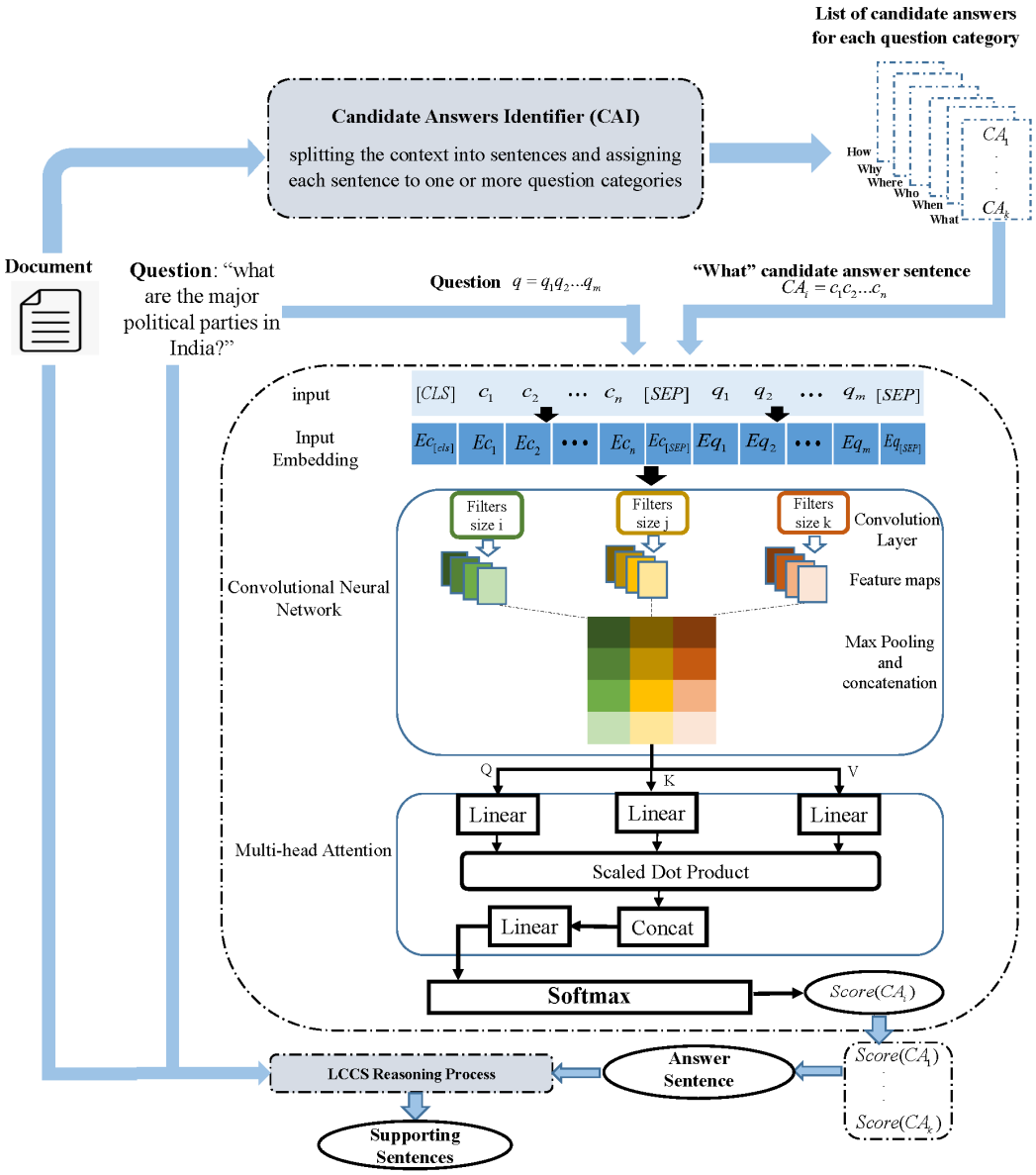

FIGURE 2 The overall framework of our Multi-hop QA model for an example "What" question. Answer sentence and supporting sentences are concatenated according to their original indexes in the given document for generating the question-driven extractive summary.

Convolutional Neural Network

CNNs are capable of learning and extracting significant n-gram features from the input text in order to generate a useful semantic representation for the subsequent tasks[64]. The convolution layer comprises of many convolution filters (also called kernel). For a sentence with n words, c_i is generated by applying the filter $\omega \in R^{ld}$ on a window of l words where l is the filter size.

Here, $e_i \in R^d$ is the word embedding for the i^{th} word in the sentence where the word embedding dimension is d, f is a non-linear activation function, and b is the bias term.

$$c_i = f(e_{i:i+l-1}.\omega^T + b) \tag{1}$$

Using the same weights, the filter ω slides across the full sentence embedding matrix to construct the feature map $c = [c_1, c_2, ..., c_i, ..., c_{n-l+1}]$. Proposing a maximum pooling method after the convolution layer diminishes the output's dimension

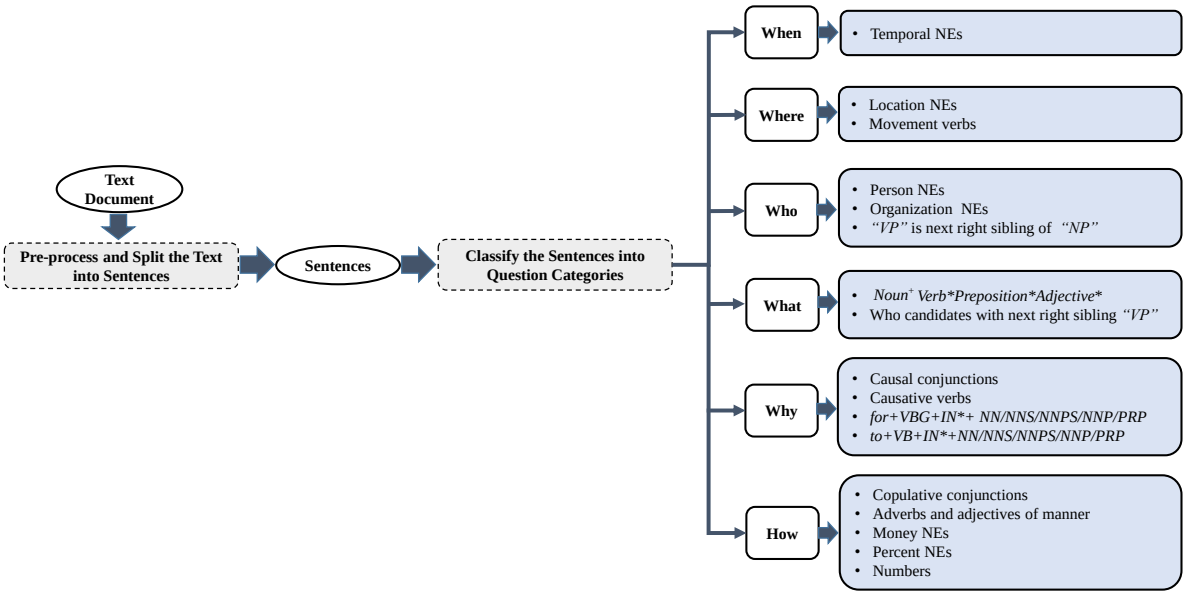

FIGURE 3 The Candidate Answer Identifier (CAI) module for identifying the candidate answer sentences for each question category (What, Where, When, Why, Who, How).

and gives us low dimension dominant features. All sampled feature maps generated by max pooling layer are combined into $\hat{C} = [\hat{c_1}, \hat{c_2}, \cdots, \hat{c_t}]$ as output of CNN ($\hat{c} = max\{c\}$).

In our model, CNN extracts key local features from the aggregated question and candidate answer sentence. The feature vectors are concatenated to construct the matrix Y as the multi-head attention layer input and global feature matrix.

Multi-head attention

The conventional attention mechanism is confined to acquiring attention information from a single level. Multiple linear transformations are performed to the input feature matrix in the multi-head attention mechanism to learn the attention representation of the text for obtaining more comprehensive semantic information[65]. We employed a multi-head attention comprising multiple self-attention mechanism (shown in Fig. 4) to assess the semantic connection between the key features of the question and candidate answer sentence for determining the relevance score. The query matrix (Q), the key matrix (K), and the value matrix (V) in self-attention mechanism are initiated with the matrix Y, the CNN layer's output.

$$Q = K = V = Y \tag{2}$$

Scaled Dot-product Attention (SDA) is one of the key concepts in the self-attention mechanism. To avoid an excessively large dot product, the dot product of Q and K is divided by $\sqrt{d_k}$ (d_k is the matrix K dimension). Multiplication by matrix V is performed to capture the expression of attention after the normalization by Softmax.

$$SDA(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

Using different parameters W_i^Q, W_i^K, W_i^V to perform linear transformation is the core idea of the multi-head attention mechanism. Applying the SDA on the linear transformation results is demonstrated by $head_i$, as shown in (4).

$$head_i = SDA(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

Concatenating the computed results $head_1$ to $head_h$ creates a matrix that is multiplied by the parameter W to complete the final linear transformation. H is the attention value of the entire sentence, depicted in (5), where h is the number of heads in the

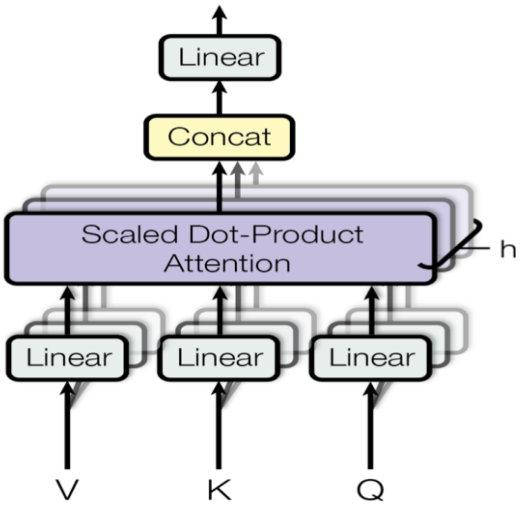

FIGURE 4 Multi-head attention structure

multi-head attention mechanism.

$$H = MultiHead(Q, K, V) = Concat(head_1, \cdots, head_h)W \tag{5}$$

We perform average pooling to derive the feature vector f for integrated q and CA_i from the output matrix of the multi-head attention layer H and then feed f into the final softmax layer through the fully connected layer. The candidate answer score is defined based on the answer selection task where two labels are used to show the answer sentence. The $Score(CA_i)$ is only calculated for label 1 for all the candidate answer sentences. The candidate with the highest score is adopted as the answer sentence for the question q. Here, C indicates the label, w_c is the weight matrix, and the bias is b_c.

$$Score(CA_i) = P(C = 1 \mid CA_i, q) \tag{6}$$

$$P(C \mid CA_i, q) = softmax(w_c f + b_c) \tag{7}$$

3.1.3 | LCCS Reasoning Process

To tackle question-driven extractive summarization, the content selection process is not only determined by answer sentence selection to the given question. It also necessitates human-like reasoning for considering the content interrelationships thoroughly and meticulously across the whole document text. In other words, if we solely focus on the answer sentence for the given question, the resulting summary is likely to miss vital information. We have proposed a reasoning process based on ***L***exical ***C***overage and ***C***ontextualized ***S***imilarity for selecting justification sentences (LCCS reasoning process). We consider all the sentences in the document (D) as the candidate justifications sentences(JC_i), and those candidates that are closest to the question (q), answer sentence (AS), and selected justification sentences (JS_i) in the embedding space are selected. We utilized pre-trained BERT for generating the contextualized embedding for the candidate sentences, question, and AS, then the cosine similarity is calculated to generate a contextualized similarity score. Also, we measure the lexical coverage of the candidates with the q, AS, and JS_i keywords (unique terms) in 8 ($X = q$,$X = AS$,$X = JS_i$).

$$C(X, JC_i) = \frac{\mid t(X) \cap t(JC_i) \mid}{max(\mid t(X) \mid, \mid t(JC_i) \mid} \tag{8}$$

$\mid t(X) \cap t(JC_i) \mid$ is the size of common terms in X and JC_i and $\mid t(X) \mid$, $\mid t(JC_i) \mid$ are the size of unique terms of X and JC_i.

3.2 | Question-driven Abstractive Model

We have proposed a paraphrase framework to transform the generated extractive summary to an abstractive summary. The input to this novel model is the extractive summary that we have obtained above. The details of this framework are presented in the

Algorithm 1 Reasoning Process

Input: Question (q), Document (D), Answer Sentence (AS), size of justification set (J-num)
Output: Set of justification sentences (JS-list) with size J-num
k=1
while (k <=J-num) **do**
  for sentence(JC) **in** D **do**
    ASq-score=C(AS,JC)+C(q,JC)+CosSimilarity(AS,JC)+CosSimilarity(q,JC)
    if ($k > 1$) **then**
      JS-score=$\sum_{i=1}^{|JS-list|} C(JS_i, JC) + CosSimilarity(JS_i, JC)$
    else
      JS-score=0
    end if
    Score(JC)=ASq-score+JS-score
  end for
  return JS=(JC with highest score)
  JS-list.Add(JS)
end while
return JS-list

following sections, and the trained paraphrase model is used for abstractive summary generation. Since our abstractive text summarizer is based on paraphrasing paradigm where the main goal is to effectively generate relevant abstractive summaries, we have found that GAN is a suitable framework to handle this task because of its ability to generate new samples.

3.2.1 | Paraphrasing for Question-driven Abstractive Summary Generation

We begin by defining two sequences of tokens $X_{1:n} = \{x_1, ..., x_n\}$ and $Y_{1:T} = \{y_1, ..., y_T\}$, where the sequence X represents an input sequence and Y represents a paraphrase. We have designed a GAN model, depicted in Fig. 5, for generating paraphrases. To this end, we have G_θ and D_ϕ to be a θ parameterized generator and a ϕ parameterized discriminator. We train G_θ to generate a sequence of tokens $\hat{Y}_{1:T} = \{\hat{y}_1, ..., \hat{y}_t\}$ that is similar to Y for the given X. We train D_ϕ to discriminate between Y and $\hat{Y}$ for input X. In the following sections, we will call X, Y, and $\hat{Y}$ as input sentence, target sentence, and generated sentence, respectively.

Generator: Generator is an encoder-decoder model based on transformers. It consists of an encoder and a decoder that are both stacks of residual attention blocks. The transformer-based encoder-decoder models process the input sequence $X_{1:n}$ of variable length n with residual attention blocks without performing a recurrent structure, which is their main advantage and innovation. Transformer-based encoder-decoders are extremely parallelizable since they don't depend on a recurrent structure, which makes them more computationally efficient on modern hardware compared to RNN-based encoder-decoder models. The transformer-based encoder encodes the input sequence $X_{1:n}$ to a sequence of hidden states and the transformer-based decoder models the conditional probability distribution of the $\hat{Y}$ sequence given the sequence of encoded hidden states from the encoder.

Discriminator: The architecture of discriminator is similar to the generator, a transformer-based encoder-decoder model that accepts X as encoder inputs, and Y (either $\hat{Y}$ or Y) as decoder input. Rather of computing a scalar as the ultimate discriminator score $D(X, \hat{Y})$, we employ a stepwise evaluation[34]. After reading the input sentence X and a portion of the output sequence $\hat{Y}_{1:t}$, the discriminator creates a scalar R_t. The ultimate discriminator score for the entire created sentence is the sum of all the scalars $R_{1:T}$ throughout the length T of the generated sequence.

$$D(X, \hat{Y}) = \frac{1}{T} \sum_{t=1}^{T} R_t \tag{9}$$

Training

At each generation step, the discriminator is customized to automatically allocate scores measuring the quality of each subsequence. Stepwise evaluation has substantially lower computational costs than MCTS, and the discriminator estimates instantaneous rewards by leveraging the idea of Q-learning and calculating state-action values without conducting tree search.

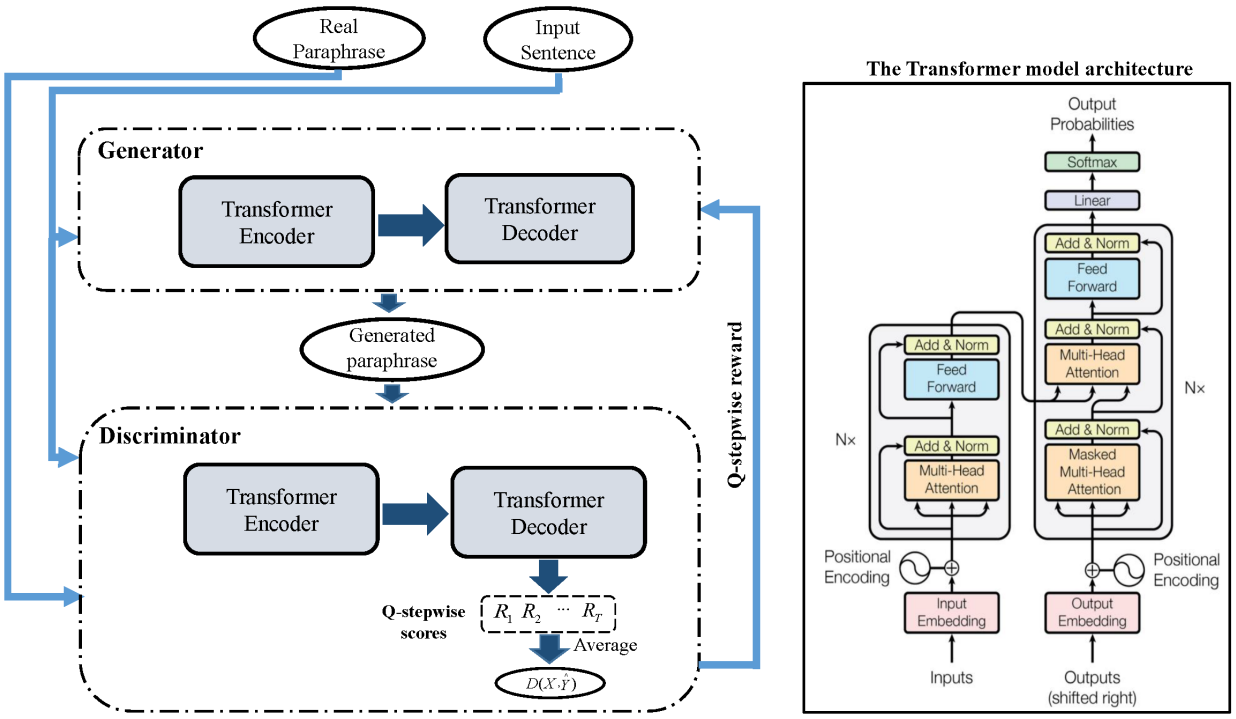

FIGURE 5 The illustration of the proposed GAN for paraphrasing

$$Q(s_t, \hat{y}_t) = \underset{z \sim P_G(.|x, \hat{y}_{1:t})}{E}[D(x, \hat{y}_{1:t}, z)] \tag{10}$$

$$R_t = Q(s_t, \hat{y}_t) \tag{11}$$

The current generator creates word sequence z with input X and generated prefix $\hat{Y}_{1:t}$.Thus, the anticipated return value of all the responses with the same prefix $\hat{y}_{1:t}$ is the state-action value $Q(s_t, \hat{y}_t)$. $s_t = (X, y_{1:t-1})$ and y_t are discrete tokens which are the inputs of the Q-function. A Kronecker delta function (or a sharp distribution) can be used for P_G in which all probabilities are zero except for the chosen sample. By this stepwise method a step dependent value, R_t, is calculated for each generation step which we call it Q-stepwise reward.

We design an approach for estimating R_t value for generator while training the discriminator. For predicting the expected value $V(s_t)$, we train a value network V that has the same structure as discriminator. The value network is trained to approximate the predicted R_t for every previous states s_t. As a result, the discriminator D_ϕ receives a pair of sentences and generates a score for each step. D_ϕ acquires knowledge using the following function:

$$J(\phi) = -log D_\phi(X, Y) - log(1 - D_\phi(X, Y)) \tag{12}$$

We train G with a stepwise evaluation technique, the objective function $J(G_\theta)$ of G_θ is:

$$J(\theta) = \sum_{t=1}^{T} R_t \nabla log P_G(y_t \mid x, y_{1:t-1}) \tag{13}$$

As the first step, we use real data to pre-train G_θ using the maximum likelihood. We also apply supervised learning to pre-train D_ϕ using pairs composed of real and created data. Then we begin several rounds of adversarial training. First, we use real samples to train G_θ using (13). We use G_θ to output a generated sample for each input sentence once we have updated the settings. As a result, D_ϕ is fed a well-balanced set of real and fake (created) pairs. Finally, we use 12 to train D_ϕ.

Algorithm 2 Training the paraphrasing model

Result: Trained G_θ
Pre-train G_θ
Generate samples using G_θ
Pre-train D_ϕ with fake and real pairs
for n rounds **do**
  for $i = 1$ to G-iteration **do**
    Sample X from real data
    Generate a sequence $\hat{Y}$ using G_θ
    Calculate R for each sequence step
    Update G_θ using equation 13
  end for
  for $j = 1$ to D-iteration **do**
    Sample (X, Y) from real data
    Sample $(X, \hat{Y})$ using G_θ
    Update D_ϕ using equation 12
  end for
end for

4 | EXPERIMENTS AND RESULTS

In this section, we present our detailed experimental study. Our goal through experiments is to demonstrate the performance of our model compared to different strong comparative models. As we use open-domain multi-hop QA system and a paraphrase generation model for producing abstractive summaries, we do need to train and evaluate them carefully due to their direct effect on summarization quality. To this end, we used three different sets of datasets and conducted an ablation analysis to evaluate our model's two components and the full question-driven abstractive text summarizer. The ablation analysis demonstrates that each of the components can produce reliable results and can be independently used. We have evaluated our approach in three different stages:

- We evaluate the proposed open-domain multi-hop QA system performance in section 4.1.
- We evaluate the paraphrase model performance in section 4.2.
- The full text summarization model is evaluated in the section 4.3.

At each section we described the relevant datasets, evaluation metrics and baseline methods for each stage.

Hyperparameter settings

We used Stanford CoreNLP[66] and settings provided in[63] for document analysis and candidate answers selection in CAI module. We have utilized the Answer Sentence Natural Questions (ASNQ)[67] derived from the Google Natural Questions (NQ) dataset[68] for training the the CNN and multi-head attention based answer selector component. ASNQ dataset contains the question, candidate answer pairs with labels, in each pair if the candidate sentence contains answer the label is 1 otherwise the label is 0. For token embeddings, we utilized the pre-trained ALBERT basic model, which consists of 12 Transformer blocks with 12 self-attention heads and a hidden size of 768. There is no mathematical procedure for determining the hyperparameters' optimal values in order to acquire the best model performance. As a result, we employed tools for tuning the model hyperparameters automatically. We optimised the hyperparameters' value using the Ray Tune Python library[1] with Hyperband algorithm[69]. Hyperband is a variation of random search employing some explore-exploit theory to determine the optimal time allocation for each configuration. Number of filters, filter size, batch size, and learning rate were optimized for training the answer selector component and their optimal values are as follows: 100,{2,3,4}, 64, 1e-5. We limited the sequence length to 128 tokens for ALBERT. We updated the parameters using the Adam optimization technique[70]. We calculated the loss using the cross-entropy

[1] https://docs.ray.io/en/latest/tune/index.html

loss function. We employed early stopping on the loss value on the development set and the maximum number of epochs is set to 10. We have utilized the pre-trained BERT basic model for generating the sentence embedding for calculating the cosine similarity in the reasoning process. The input representation for our paraphrase model is the pre-trained wordpiece embeddings from ALBERT. For training the parapharse model, we trained the model for 10 epochs by Q-stepwise evaluation method after pre-training the generator by MLE. The discriminator is pre-trained on the generated samples from the pre-trained generator and real data. We adopted Adam optimization algorithm to pre-train the generator and train the discriminator. The optimal learning rates for G_θ, D_ϕ are $2e-6$, $5e-6$ calculated by Hyperopt algorithm. Hyperopt determines the optimal batch size 32 and 64 for Quora (100K, 150K) and MSCOCO[71] datasets to feed our generator and discriminator, and we performed 20 rounds of adversarial training.

4.1 | Open-domain multi-hop QA

We used MultiRC dataset for evaluating the proposed open-domain multi-hop QA model. Multi-sentence reading comprehension (MultiRC) is a reading comprehension dataset administered via a multiple-choice QA task[72]. Each question is based on a paragraph that comprises the question's gold justification sentences.

We used $F1_m$, $F1_a$, and EM evaluation metrics introduced in[72]. Table 2 summarizes the experimental results for open-domain multi-hop QA and four baseline methods which are described below.

WAIR[73] utilizes the alignment IR approach[74] to retrieve justification sentences and a RoBERTa binary classifier for answer selection. The WAIR technique, in two iterations, reduces the weights of question terms that have already been addressed by previously retrieved sentences and increases the weights of reformulated question terms that have not yet been covered. The second iteration reranks the clusters of evidence sentences using a regression task, with each sentence cluster allocated an F1 score generated from the gold annotated evidence sentences.

AIR[75] discovers justification sentences by an unsupervised strategy based on GloVe embeddings and an alignment model. To choose answers, a RoBERTa binary classifier is utilized. The question and candidate answer text are used to initiate the query. AIR adjusts its query after each repetition to focus on the missing information in the current set of justifications. The alignment approach computes the cosine similarity between each token's word embeddings in the query and the provided text sentence, resulting in a matrix of cosine similarity scores.

ROCC[76] presented an unsupervised technique for maximising the relevance of selected sentences, minimising the overlap between selected facts, and maximising both question and answer coverage. The relevance, coverage, and overlap scores of candidate justification sets are calculated. They used BERT as a binary classifier to choose answers.

Multee[77] presented models of entailment for multi-hop QA composing a relevance module and multi-layer aggregation module. Both modules make use of ESIM[78], a recently developed sentence-level entailment model that has been trained on the SNLI and MultiNLI datasets.

We have reported the results for the baseline methods from their paper. It is evident that WAIR outperformed other baselines since it introduced several attention and embedding-based analyses. It demonstrates that by combining retrieval and reranking techniques, it is possible to acquire the compositional knowledge necessary for multi-hop reasoning. AIR is the second-best baseline and outperformed ROCC and Multee due to the iterative method used to reformulate queries and focus on words not covered by existing justifications. AIR is an unsupervised alignment technique that uses only GloVe embeddings to soft-align questions and answers with justification sentences. ROCC outperformed Multee because it is an unsupervised strategy that utilizes a BERT answer classifier and three scoring functions to rank candidate reason sets. In compared to Multee's entailment technique, the ranking functions improve ROCC performance by increasing the relevance of the selected sentences and decreasing lexical overlap between the selected facts. Our model (CNN-Att-MhopQA) outperformed all baselines since it investigates the the semantic correlations between the features extracted from the question and relevant sentences in document. The semantic correlations between features are obtained by applying CNN and multi-head attention to the combined representation of the question and candidate answer sentences. Also, the reasoning process based on lexical coverage and BERT embedding is a complement to answer selector module for selecting justification sentences. We have added the results of the proposed model with self-attention and compared the performance with multi-head attention. As it is shown, the multi-head attention variant performs better due to the less number of layers and training stability in comparison to self-attention variant. Fig. 6 shows the CNN-Att-MhopQA performance on each question category for MultiRC dataset. The multi-head attention variant outperformed the self-attention variant on all question categories since multi-level attention information is acquired.

TABLE 2 $F1_m$, $F1_a$, and EM score for our method and open-domain Multi-hop QA baseline methods on MultiRC dataset.

Model	MultiRC dataset		
	$F1_m$	$F1_a$	EM
WAIR[73]	79.5	76.5	35.4
AIR[75]	79.0	76.4	36.3
ROCC[76]	73.8	70.6	26.1
Multee[77]	71.7	68.3	-
CNN-Att-MhopQA(ours)	82.2	79.8	40.3
CNN-Att-MhopQA (with self-attention)	80.7	77.9	37.6

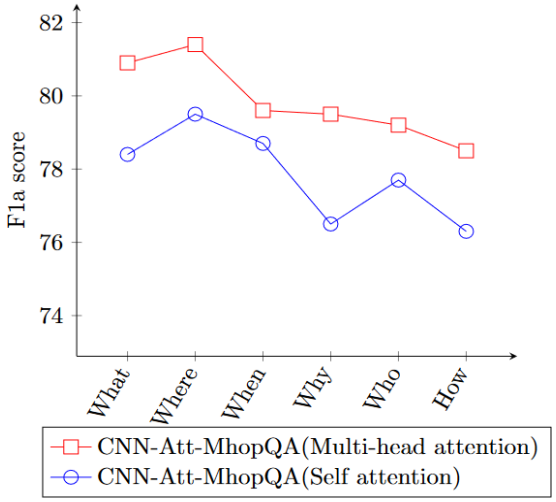

FIGURE 6 CNN-Att-MhopQA multi-head attention and self-attention variants' performance on each question category for the MultiRC dataset.

4.2 | Paraphrase generation model

At the second stage of our experiments, we implement and evaluate our paraphrase generation model (QTrans-GAN) independently to assess its capability for paraphrasing extractive summaries. We choose the two most widely used datasets, Quora[2] and MSCOCO[71] for paraphrase generation experiments.

- **Quora** dataset consists of over 400K candidate question paraphrase pairs with manually annotated labels. Two questions are paraphrasing each other only when the question pair's label is 1. We have used two different training sizes (100K and 150K) from Quora to have the same setting with baseline methods and show how the size of the dataset can affect the results of paraphrase generation.
- **MSCOCO** is a benchmark for the task of image captioning which contains over 82K training and 42K validation images, and at most five human-labeled caption are provided for each image. Similar to the previous works on paraphrase generation, different captions of the same image are considered as paraphrases.

We used some automatic metrics to evaluate QTrans-GAN framework and compare it with other methods.

- BLEU4[79] is the most widely used evaluation metric in paraphrase generation. This approach works by counting matching n-grams in the generated sentence and the reference sentence.

[2]https://www.quora.com/share/First-Quora-Dataset-Release-Question-Pairs

TABLE 3 Experimental results of paraphrase generation on Quora (with 100K and 150K training set size) and MSCOCO datasets. The results for EndtoEnd-GAN, Div-GAN, and Pen-GAN are reported from their paper.

Method	Quora-100K		Quora-150K		MSCOCO	
	BLEU4	METEOR	BLEU4	METEOR	BLEU4	METEOR
EndtoEnd-GAN[60]	41.33	28.46	43.31	28.25	42.53	32.77
Div-GAN[61]	-	-	28.49	-	20.63	-
Pen-GAN[62]	29.07	31.27	-	-	-	-
SE-GAN[34]	41.96	30.36	43.62	31.04	42.70	32.89
QTrans-GAN(ours)	**43.79**	**32.41**	**44.71**	**33.23**	**45.03**	**33.97**

- METEOR[80] metric is based on the harmonic mean of unigram precision and recall, with recall taking precedence over precision. Along with the basic precise word matching, it also offers other features which are not found in other measures, such as stemming and synonym matching. METEOR was designed to address some of the flaws in the BLEU metric while also producing a high level of correlation with human judgement at the segment or sentence level.

We employ four recent paraphrase generation approaches based on GANs as baseline methods, which are described in detail below.

EndtoEnd-GAN[60] regarded the generator (two stacked LSTMs encoder and decoder) as the stochastic policy and the output of discriminator (one LSTM) as its reward. In this way, they propagated the gradients from the discriminator to both the generator models and encoder models.

Div-GAN[61] proposed a conditional GAN-based framework consisting a GRU-based generator and a CNN-based discriminator. They adopted the policy gradient and early feedback techniques described in[32] for training.

Pen-GAN[62] utilized a Convolutional seq2seq model for both generator and discriminator. They engage the discriminator output as penalization rather than using policy gradients, and they avoid the Monte-Carlo search by proposing a global discriminator.

SE-GAN[34] proposed the stepwise evaluation for chit-chat dialogue generation using GRU encoder decoder for both generator and discriminator and estimated state-action values for each generation step by modifying the architecture of the discriminator. We have applied this approach for paraphrase generation as a baseline method.

Table 3 summarizes the experimental results for paraphrasing on Quora (with 2 training sizes, 100K and 150K) and MSCOCO datasets. We reported the results for EndtoEnd-GAN, Div-GAN, and Pen-GAN from their paper. SE-GAN outperformed on all datasets compared to other baseline methods due to employing stepwise evaluation. Div-GAN has the worst performance on Quora-150K and MSCOCO datasets because of using policy gradient. EndtoEnd-GAN and Pen-GAN are in the second and third places respectively regarding their BLEU scores; however, Pen-GAN has a better METEOR score on Quora-100K dataset. EndtoEnd-GAN outperfromed Pen-GAN because of proposing a generator based on stacked LSTMs and applying stochastic policy. Our model improved the BLEU and METEOR scores compared to all these baseline methods because of using transformers and Q-stepwise rewarding jointly in the discriminator. In detail, stacks of residual attention blocks in transformer, not relying on a recurrent structure, and reward calculation based on Q-learning for each generation step are the reasons for better performance in QTrans-GAN.

4.3 | Question-driven abstractive text summarization

In the third stage, we evaluate our hybrid summarization framework (QParaSum) using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric[81], it compares an automatically generated summary with a set of human-produced summaries. ROUGE-N measures unigrams, bigrams, trigrams, and higher-order n-grams overlap. ROUGE-L utilizes the Longest Common Subsequence method (LCS) to determine the longest matching sequence of words. We don't require a predefined n-gram length since it automatically contains the longest in-sequence common n-grams. We evaluate QParaSum model on three large-scale summarization datasets, WikiHow[82], PubMedQA[83], and MEDIQA dataset[84].

- **WikiHow** is a dataset accumulated from the WikiHow community-based QA website for abstractive text summarization task. Each sample in WikiHow dataset consists of a lengthy article, a non-factoid question, and the associated summary as the answer to the question.
- **PubMedQA** is a biomedical QA dataset derived from PubMed2 abstracts. Each sample includes a question, an article, and an abstractive answer which summarizes the context corresponding to the question.
- **MEDIQA** is a dataset comprising 156 consumer-submitted health questions, corresponding articles to these questions, and expert-written summaries of the answers.

At the final phase of our experiments, we consider four recent question-driven text summarization and three query-based[3] baseline methods for evaluating QParaSum model.

HSCM[24] presented an approach for extractive answer summarization consisting of three components. In the first and second components (Word-level and Sentence-level Compare-Aggregate), an attention operation is used to align the word-level and sentence-level information between the answer sentence and question. In the third component, Question-aware Sequential Extractor, a RNN decoder is designed to label each sentence consecutively and construct the answer summary for the target question.

MSG[22] proposed Multi-hop Selective Generator (MSG), a question-driven abstractive summarization approach that integrates multi-hop reasoning to identify the key content for assisting the answer generation. In addition to the multi-view pointer network, they introduced a multi-view coverage technique to overcome the duplication issue and generate informative and precise answers.

QPGN[25] presented a question-driven pointer-generator network that utilizes the correlation information between question-answer pairs to add substantial information when generating abstractive answer summaries. Their framework consists of four components: Bi-LSTM Encoder, seq2seq Model joint with question-aware attention, question-answer alignment with summary representations, question-driven Pointer-generator Network.

Trans[26] has studied the capability of three state-of-the-art transformers for question-driven text summarization: BART, T5, and PEGASUS in both zero-shot and few-shot learning settings for question-driven abstractive text summarization on MEDIQA dataset. T5 outperformed the others thus we consider it as a baseline method.

Div-qsum[29] introduced a typical encode-attend-decode model (based on LSTM) for query-based abstractive summarization, which first computes a vectorial representation for the document and the query, and then the decoder produces a contextual summary one word at a time.

PGRU-qsum[85] is a pointer-generator model based on GRU encoder-decoder with attention and a pointer mechanism, for generating query-based summaries.

SummerTime[86] is a comprehensive text summarizing toolkit that interfaces with libraries built for NLP researchers and provides simple-to-use APIs to users. For query-based summarization, the top-k query-relevant phrases are retrieved using TF-IDF and BM25.

Table 4 shows the experimental results for one extractive (HSCM), three abstractive question-driven summarization approaches (MSG, QPGN, Tran(T5)), and three query-based summarization approaches (Div-qsum, PGRU-qsum, SummerTime) on WikiHow, PubMedQA, MEDIQA datasets. The results for HSCM, MSG, QPGN, Tran(T5) are reported from their papers, and the results for query-based baselines are generated by using their public code for our datasets. ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE (RL) are considered to evaluate the quality of our extractive and abstractive summaries. We have included the evaluation for our question-driven extractive summary to assess the impact of the paraphrasing process for generating abstractive summaries. HSCM generated extractive answer summaries and as it is shown in Table 4 the R1, R2, and RL for our extractive summaries and other abstractive baseline methods are superior to HSCM. Employing the GloVe language model and relying on a recurrent structure decoder for generating extractive answer summaries caused this inefficiency and poor performance in HSCM. MSG achieves relatively better performance than QPGN because of incorporating multi-hop reasoning for abstractive summarization. MSG and QPGN have used the pre-trained Glove model[87] which is not a very efficient model because of the co-occurrence matrix of words that consumes a considerable amount of memory. Besides using an inefficient language model, having multi-stages of training is another problem of these baseline methods. In Trans, each language generation model (BART, T5, PEGASUS) is pre-trained with different strategies which are unclear whether these strategies are the optimal ones. The query-based baselines have poor performances compared to question-driven baselines since answer selection and justification are not considered in query-based summarization. PGRU-qsum outperformed Div-qsum and SummerTime because

[3] We used their public code to apply their model to the datasets and generate question-driven summaries

TABLE 4 Results on WikiHow, PubMedQA, and MEDIQA. The results for HSCM, QPGN, MSG, and Trans(T5) are reported from their paper.

	WikiHow			PubMedQA			MEDIQA		
Model	R1	R2	RL	R1	R2	RL	R1	R2	RL
HSCM[24]	27.84	7.75	25.85	32.34	10.07	25.98	-	-	-
QPGN[25]	28.8	9.7	27.7	34.2	12.8	28.7	-	-	-
MSG[22]	30.5	10.5	29.3	37.2	14.8	30.2	-	-	-
Trans(T5)[26]	-	-	-	-	-	-	38.56	18.52	26.00
PGRU-qsum	19.82	6.41	17.96	24.58	8.13	17.67	25.32	8.98	18.42
Div-qsum	18.56	5.10	15.81	22.67	7.55	16.80	23.56	7.08	17.67
SuumerTime	15.43	4.67	13.78	19.67	5.98	14.60	20.42	6.31	15.66
QParaSum-Extractive(ours)	**31.71**	**11.23**	**30.09**	**38.89**	**14.81**	**30.76**	**40.94**	**20.11**	**27.46**
QParaSum-Abstractive(ours)	**33.69**	**12.05**	**31.79**	**41.00**	**16.42**	**32.93**	**44.32**	**23.02**	**29.81**

of utilizing attention and pointer mechanism. Div-qsum outperformed SummerTime due to using an attention mechanism for encoding documents and queries.

Favorably QParaSum model obtains the state-of-the-art results for all three datasets with the generated extractive and abstractive summaries. The results indicate that the generated extractive summary covers the essential information for satisfying answerability, understandability, and persuasiveness measures by finding the AS and its supporting sentences using the proposed multi-hop QA system. Also, the extractive stage prunes the text (the input) for abstractive stage by removing the irrelevant and redundant information regarding the question. It was shown that the idea of exploiting an appropriate paraphrasing model for transforming the extractive summaries to abstractive is feasible since for all the three datasets the abstractive summaries obtain higher R1, R2, and RL. The parapharse model makes our hybrid summarizer capable to generate high quality abstractive summaries that are more close to the human generated ones. In Table 5, we show a practical example of our framework outputs for an instance from MEDIQA dataset. At first stage, the AS is selected by the proposed answer selector module based on CNN and multi-head attention and then IS1 is detected by the proposed LCCS reasoning method as the most semantically relevant sentence to AS. IS2 and IS3 are selected as the next supporting sentences for IS1 and AS. Since the IS1 contains general and important information about the AS (it contains the symptoms, medication and surgery as the treatment for hernia), we need to find sentences explaining and extending this information. The extractive summary is constructed by concatenating AS, IS1, IS2, and IS3 with the same order that they have in the article. At the second stage, the trained paraphrase model (the generator) is used for rewriting the sentences in the extractive summary to improve it and make it more similar to the human generated summary. After generating the extractive summary, it is evident that we have tried to simulate the human action for text summarization, regenerating and rewriting the sentences regarding to their understanding from text, using a paraphrase model on the extractive summary. To evaluate the generated abstractive summary and compare its quality to the extractive summary, we have used the human generated summary (gold summary) and R1, R2, RL metrics to demonstrate whether our abstractive summary is similar to the gold summary. The generated abstractive summary obtained higher R1, R2, and RL, and it shows that our abstractive summary has more in common sequences of words with the gold summary. Fig. 7 shows the average of R1, R2, and RL scores for QParaSum-Abstractive model across all datasets with different question lengths. It is evident that the model performance is not impacted by the length of the input question since an insignificant performance degradation (R1, R2, RL) is observed when the question length increases. However, for generating shorter abstractive summaries, merging the sentences could be considered during the paraphrase stage, which is the goal for our future work.

4.4 | Discussion

We have proposed a novel paraphrasing model for generating abstractive summaries. To begin with, the most relevant pieces of information are selected from the text regarding the target question for constructing the question-driven extractive summary. The trained paraphrase generation model is applied to the selected sentences to rewrite them and generate the abstractive summary. The results show that the generated abstractive summary is closer to the gold summary compared to the generated extractive

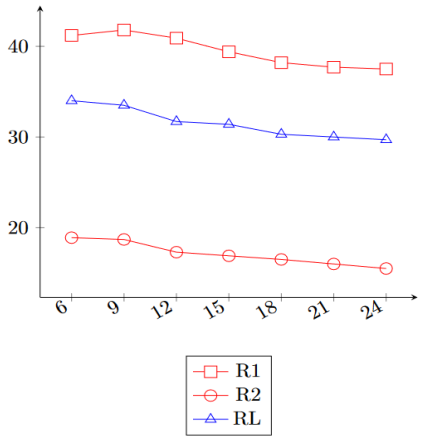

FIGURE 7 QParaSum-Abstractive performance (average R1, R2, RL across all datasets) with different question lengths.

TABLE 5 An example from MEDIQA dataset for extractive and abstractive summaries generated by our framework that are evaluated by the gold summary.

Question: I have an hernia I would love to take care of it ASAP I was wondering if you guys could help and tell me what should I do?
Article: Hiatal hernia (Treatment): The majority of patients who have a hiatal hernia will exhibit no signs or symptoms and will not require treatment. If you have persistent heartburn or acid reflux, you may require medication or surgery. If you suffer from heartburn or acid reflux, your doctor may prescribe the following medications:- Antacids that act as a buffer for stomach acid. Anti-acid medications such as Mylanta, Rolaids, and Tums may give immediate relief. Certain antacids may have adverse effects such as diarrhoea or kidney problems if used in excess.- Medications that inhibit acid production. Cimetidine (Tagamet), famotidine (Pepcid), nizatidine (Axid), and ranitidine are all H-2 receptor antagonists (Zantac). Prescriptions are required for stronger versions. - Anti-acid medications that aid in the healing of the esophagus. Proton pump inhibitors are more effective acid blockers than H-2 receptor antagonists, and they allow a longer time for injured esophageal tissue to repair. Lansoprazole (Prevacid 24HR) and omeprazole are two proton pump inhibitors available over-the-counter (Prilosec, Zegerid). Surgery is normally reserved for those who are unable to control their heartburn or acid reflux with medicines or who have problems such as significant inflammation or esophageal constriction. Surgery to repair a hiatal ...
Question-driven Extractive Summary: (Answer Sentence) Hiatal hernia (Treatment): The majority of patients who have a hiatal hernia will exhibit no signs or symptoms and will not require treatment. (IS1) If you have persistent heartburn or acid reflux, you may require medication or surgery. (IS2) If you suffer from heartburn or acid reflux, your doctor may prescribe the following medications:- Antacids that act as a buffer for stomach acid. (IS3) Surgery is normally reserved for those who are unable to control their heartburn or acid reflux with medicines or who have problems such as significant inflammation or esophageal constriction.
Question-driven Abstractive Summary: Hiatal hernia (Treatment): Most individuals with a hiatal hernia don't have any signs or symptoms and will not require treatment. If you have signs like repetitive acid reflux and heartburn, you may require medication or surgery. Your doctor may recommend Antiacids to neutralize stomach acid if you experience acid reflux and heartburn. Surgery is recommended if the medications do not help the individual to soothe acid reflux and heartburn, or have complexities like serious inflammation or narrowing of the esophagus.
Gold Summary: If a hiatal hernia does not have any symptoms, it won't require treatment. If the hernia causes heartburn and acid reflux, your doctor may recommend antacids. If the medications do not help or hiatal hernia causes inflammation or narrowing of the esophagus, your doctor might recommend surgery.
Extractive Summary Evaluation:{R1:0.406, R2: 0.148, RL: 0.260} **Abstractive Summary Evaluation**:{R1: 0.589, R2: 0.314, RL: 0.434}

summary. The reason for this quality improvement in the abstractive summary is empowering the GAN with transformers and Q-learning stepwise evaluation for paraphrase generation which is applied on the extractive summary generated by an open-domain multi-hop QA system. We have explored the previous question-driven text summarization approaches shortcomings and considered them while proposing our model. HSCM[24], MSG[22], and QPGN[25] have used the pre-trained Glove model which is not efficient because of the co-occurrence matrix of words that takes a lot of memory for storage. Besides, having multiple

steps of training is another problem in these approaches. In Trans[26] each language generation model (BART, T5, PEGASUS) is pre-trained with different strategies which are unclear whether these strategies are the optimal ones. We have designed two stages of training which are done once for open-domain question-driven text summarization while it could be even tuned for a specific domain. In our model, the extractive model is designed for producing extractive summaries with four sentences, while the average length of the gold summaries is four sentences, and calculating the optimal summary length for each instance is considered as one of our future works. In the paraphrase generation stage, we have proposed a sentence-level paraphrase model that processes the extractive summary sentences one by one and generates an abstractive summary which is more close to the gold summary as the results show in section 4.3. Although considering the complete extractive summary is more desirable, proposing a framework capable of paraphrasing and merging some sentences at the same time will be considered in our future work. In other words, generating the correct link between the sentences and condensing them to improve the summary coherence is the complementary idea for this paper which we will study in the future.

5 | CONCLUSION

We propose a novel open-domain question-driven hybrid text summarization method incorporating an open-domain multi-hop QA system for extractive summarization and a paraphrase model to regenerate the extractive summaries and constructing the abstractive ones. We showed that the proposed paraphrase model based on GANs and transformers with Q-learning stepwise evaluation can transform the extractive summaries into abstractive. To the best of our knowledge, this is the first work that employs a paraphrase generation model to generate abstractive summaries. We have evaluated our results on WikiHow, PubMedQA, and MEDIQA datasets which are appropriate for the question-driven summarization problems. We have compared our results to several baseline methods, and we can conclude that our hybrid framework is more effective for question-driven text summarization.

6 | ACKNOWLEDGEMENTS

Research reported in this manuscript is sponsored by Computer Science and Electronic Engineering Department at University of Essex, and BT group.

References

1. Kryściński Wojciech, Paulus Romain, Xiong Caiming, Socher Richard. Improving Abstraction in Text Summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing:1808–1817; 2018.

2. El-Kassas Wafaa S, Salama Cherif R, Rafea Ahmed A, Mohamed Hoda K. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications.* 2021;165:113679.

3. Suleiman Dima, Awajan Arafat. Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. *Mathematical Problems in Engineering.* 2020;2020:1–29.

4. Li Piji, Lam Wai, Bing Lidong, Wang Zihao. Deep Recurrent Generative Decoder for Abstractive Text Summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing:2091–2100; 2017.

5. Kirmani Mahira, Hakak Nida Manzoor, Mohd Mudasir, Mohd Mohsin. Hybrid text summarization: a survey. In: Soft Computing: Theories and Applications. Springer 2019 (pp. 63–73).

6. AKSENOV DMITRII. Abstractive text summarization with neural sequence-to-sequence models. Master's thesis2020.

7. Zhang Jingqing, Zhao Yao, Saleh Mohammad, Liu Peter. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning:11328–11339PMLR; 2020.

8. Wang Li, Yao Junlin, Tao Yunzhe, Zhong Li, Liu Wei, Du Qiang. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence:4453–4460; 2018.

9. Song Kaiqiang, Wang Bingqing, Feng Zhe, Liu Ren, Liu Fei. Controlling the amount of verbatim copying in abstractive summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence:8902–8909; 2020.

10. Shi Tian, Keneshloo Yaser, Ramakrishnan Naren, Reddy Chandan K. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science.* 2021;2(1):1–37.

11. Kouris Panagiotis, Alexandridis Georgios, Stafylopatis Andreas. Abstractive text summarization: enhancing sequence to sequence models using word sense disambiguation and semantic content generalization. *Computational Linguistics.* 2021;:1–41.

12. Cao Ziqiang, Li Wenjie, Wei Furu, Li Sujian, others . Retrieve, rerank and rewrite: Soft template based neural summarization. In: Association for Computational Linguistics (ACL); 2018.

13. Yang Min, Qu Qiang, Tu Wenting, Shen Ying, Zhao Zhou, Chen Xiaojun. Exploring human-like reading strategy for abstractive text summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence:7362–7369; 2019.

14. Kanapala Ambedkar, Pal Sukomal, Pamula Rajendra. Text summarization from legal documents: a survey. *Artificial Intelligence Review.* 2019;51(3):371–402.

15. Abdelaleem Nadeen M, Kader HM Abdal, Salem Rashed. A Brief Survey on Text Summarization Techniques. *IJ of Electronics and Information Engineering.* 2019;10(2):103–116.

16. Kornilova Anastassia, Eidelman Vladimir. BillSum: A Corpus for Automatic Summarization of US Legislation. In: Proceedings of the 2nd Workshop on New Frontiers in Summarization:48–56Association for Computational Linguistics; 2019; Hong Kong, China.

17. Kieuvongngam Virapat, Tan Bowen, Niu Yiming. Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2. *arXiv e-prints.* 2020;:arXiv–2006.

18. Afzal Muhammad, Alam Fakhare, Malik Khalid Mahmood, Malik Ghaus M. Clinical Context–Aware Biomedical Text Summarization Using Deep Neural Network: Model Development and Validation. *Journal of medical Internet research.* 2020;22(10):e19810.

19. Gharebagh Sajad Sotudeh, Goharian Nazli, Filice Ross. Attend to Medical Ontologies: Content Selection for Clinical Abstractive Summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:1899–1905; 2020.

20. Abdi Asad, Shamsuddin Siti Mariyam, Aliguliyev Ramiz M. QMOS: Query-based multi-documents opinion-oriented summarization. *Information Processing & Management.* 2018;54(2):318–338.

21. Adhikari Surabhi, others . Nlp based machine learning approaches for text summarization. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC):535–538IEEE; 2020.

22. Deng Yang, Zhang Wenxuan, Lam Wai. Multi-hop Inference for Question-driven Summarization. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP):6734–6744; 2020.

23. Song Hongya, Ren Zhaochun, Liang Shangsong, Li Piji, Ma Jun, Rijke Maarten. Summarizing answers in non-factoid community question-answering. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining:405–414; 2017.

24. Deng Yang, Zhang Wenxuan, Li Yaliang, Yang Min, Lam Wai, Shen Ying. Bridging Hierarchical and Sequential Context Modeling for Question-driven Extractive Answer Summarization. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval:1693–1696; 2020.

25. Deng Yang, Lam Wai, Xie Yuexiang, et al. Joint learning of answer selection and answer summary generation in community question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence:7651–7658; 2020.

26. Goodwin Travis R, Savery Max E, Demner-Fushman Dina. Flight of the PEGASUS? Comparing Transformers on Few-Shot and Zero-Shot Multi-document Abstractive Summarization. In: Proceedings of COLING. International Conference on Computational Linguistics:5640NIH Public Access; 2020.

27. Afsharizadeh Mahsa, Ebrahimpour-Komleh Hossein, Bagheri Ayoub. Query-oriented text summarization using sentence extraction technique. In: 2018 4th international conference on web research (ICWR):128–132IEEE; 2018.

28. Van Lierde Hadrien, Chow Tommy WS. Query-oriented text summarization based on hypergraph transversals. *Information Processing & Management.* 2019;56(4):1317–1338.

29. Nema Preksha, Khapra Mitesh M, Laha Anirban, Ravindran Balaraman. Diversity driven attention model for query-based abstractive summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers):1063–1072; 2017.

30. Ishigaki Tatsuya, Huang Hen-Hsen, Takamura Hiroya, Chen Hsin-Hsi, Okumura Manabu. Neural query-biased abstractive summarization using copying mechanism. *Advances in Information Retrieval.* 2020;12036:174.

31. Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, et al. Generative adversarial networks. *Communications of the ACM.* 2020;63(11):139–144.

32. Yu Lantao, Zhang Weinan, Wang Jun, Yu Yong. Seqgan: Sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI conference on artificial intelligence; 2017.

33. Sutton Richard S, Barto Andrew G. *Reinforcement learning: An introduction*. MIT press; 2018.

34. Tuan Yi-Lin, Lee Hung-Yi. Improving conditional sequence generative adversarial networks by stepwise evaluation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing.* 2019;27(4):788–798.

35. Watkins Christopher JCH, Dayan Peter. Q-learning. *Machine learning.* 1992;8(3-4):279–292.

36. Wu Qingyang, Li Lei, Yu Zhou. TextGAIL: Generative Adversarial Imitation Learning for Text Generation. In: Proceedings of the AAAI Conference on Artificial Intelligence:14067–14075; 2021.

37. Radford Alec, Wu Jeffrey, Child Rewon, et al. Language models are unsupervised multitask learners. *OpenAI blog.* 2019;1(8):9.

38. Zhang Chao, Xiong Caiquan, Wang Lingyun. A Research on Generative Adversarial Networks Applied to Text Generation. In: 2019 14th International Conference on Computer Science Education (ICCSE):913-917; 2019.

39. Konda Vijay R, Tsitsiklis John N. Actor-critic algorithms. In: Advances in neural information processing systems:1008–1014; 2000.

40. Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.* 2018;.

41. Lan Zhenzhong, Chen Mingda, Goodman Sebastian, Gimpel Kevin, Sharma Piyush, Soricut Radu. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In: International Conference on Learning Representations; 2019.

42. Clark Kevin, Luong Minh-Thang, Le Quoc V, Manning Christopher D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555.* 2020;.

43. Li Weikang, Zhang Xingxing, Wu Yunfang, Wei Furu, Zhou Ming. Document-Based Question Answering Improves Query-Focused Multi-document Summarization. In: CCF International Conference on Natural Language Processing and Chinese Computing:41–52Springer; 2019.

44. Zhao Mingjun, Yan Shengli, Liu Bang, et al. QBSUM: A large-scale query-based document summarization dataset from real-world applications. *Computer Speech & Language.* 2021;66:101166.

45. Liu Linqing, Lu Yao, Yang Min, Qu Qiang, Zhu Jia, Li Hongyan. Generative adversarial network for abstractive text summarization. In: Thirty-second AAAI conference on artificial intelligence; 2018.

46. Scialom Thomas, Dray Paul-Alexis, Lamprier Sylvain, Piwowarski Benjamin, Staiano Jacopo. Discriminative adversarial search for abstractive summarization. In: International Conference on Machine Learning:8555–8564PMLR; 2020.

47. Dong Li, Yang Nan, Wang Wenhui, et al. Unified language model pre-training for natural language understanding and generation. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems:13063–13075; 2019.

48. Rekabdar Banafsheh, Mousas Christos, Gupta Bidyut. Generative adversarial network with policy gradient for text summarization. In: 2019 IEEE 13th international conference on semantic computing (ICSC):204–207IEEE; 2019.

49. Dang Nobel, Khanna Ashish, Allugunti Viswanatha Reddy. TS-GAN with Policy Gradient for Text Summarization. In: Data Analytics and Management. Springer 2021 (pp. 843–851).

50. Wang Shuai, Zhao Xiang, Li Bo, Ge Bin, Tang Daquan. Integrating extractive and abstractive models for long text summarization. In: 2017 IEEE International Congress on Big Data (BigData Congress):305–312IEEE; 2017.

51. Bhat Iram Khurshid, Mohd Mudasir, Hashmy Rana. Sumitup: A hybrid single-document text summarizer. In: Soft computing: Theories and applications. Springer 2018 (pp. 619–634).

52. Subramanian Sandeep, Li Raymond, Pilault Jonathan, Pal Christopher. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186.* 2019;.

53. Chen Yangbin, Ma Yun, Mao Xudong, Li Qing. Multi-task learning for abstractive and extractive summarization. *Data Science and Engineering.* 2019;4(1):14–23.

54. Jin Hanqi, Wang Tianming, Wan Xiaojun. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:6244–6254; 2020.

55. Gupta Ankush, Agarwal Arvind, Singh Prawaan, Rai Piyush. A deep generative framework for paraphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2018.

56. Li Zichao, Jiang Xin, Shang Lifeng, Liu Qun. Decomposable Neural Paraphrase Generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics:3403–3414; 2019.

57. Fu Yao, Feng Yansong, Cunningham John P. Paraphrase Generation with Latent Bag of Words. *Advances in Neural Information Processing Systems.* 2019;32:13645–13656.

58. Siddique AB, Oymak Samet, Hristidis Vagelis. Unsupervised paraphrasing via deep reinforcement learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining:1800–1809; 2020.

59. Liu Xianggen, Mou Lili, Meng Fandong, Zhou Hao, Zhou Jie, Song Sen. Unsupervised Paraphrasing by Simulated Annealing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:302–312; 2020.

60. Yang Qian, Huo Zhouyuan, Shen Dinghan, et al. An end-to-end generative architecture for paraphrase generation. 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):3132–3142; 2019.

61. Cao Yue, Wan Xiaojun. DivGAN: Towards diverse paraphrase generation via diversified generative adversarial network. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings:2411–2421; 2020.

62. Vizcarra Gerson, Ochoa-Luna Jose. Paraphrase Generation via Adversarial Penalizations. In: Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020):249–259; 2020.

63. Kia Mahsa Abazari, Garifullina Aygul, Kern Mathias, Chamberlain Jon, Jameel Shoaib. Adaptable Closed-Domain Question Answering Using Contextualized CNN-Attention Models and Question Expansion. *IEEE Access.* 2022;10:45080-45092.

64. Young Tom, Hazarika Devamanyu, Poria Soujanya, Cambria Erik. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine.* 2018;13(3):55–75.

65. Vaswani Ashish, Shazeer Noam, Parmar Niki, et al. Attention is all you need. In: Advances in neural information processing systems:5998–6008; 2017.

66. Manning Christopher D, Surdeanu Mihai, Bauer John, Finkel Jenny Rose, Bethard Steven, McClosky David. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations:55–60; 2014.

67. Garg Siddhant, Vu Thuy, Moschitti Alessandro. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In: Proceedings of the AAAI Conference on Artificial Intelligence:7780–7788; 2020.

68. Kwiatkowski Tom, Palomaki Jennimaria, Redfield Olivia, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics.* 2019;7:453–466.

69. Li Lisha, Jamieson Kevin, DeSalvo Giulia, Rostamizadeh Afshin, Talwalkar Ameet. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research.* 2017;18(1):6765–6816.

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

71. Lin Tsung-Yi, Maire Michael, Belongie Serge, et al. Microsoft coco: Common objects in context. In: European conference on computer vision:740–755Springer; 2014.

72. Khashabi Daniel, Chaturvedi Snigdha, Roth Michael, Upadhyay Shyam, Roth Dan. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers):252–262; 2018.

73. Yadav Vikas, Bethard Steven, Surdeanu Mihai. If You Want to Go Far Go Together: Unsupervised Joint Candidate Evidence Retrieval for Multi-hop Question Answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies:4571–4581; 2021.

74. Yadav Vikas, Bethard Steven, Surdeanu Mihai. Alignment over heterogeneous embeddings for question answering. 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):2681–2691; 2019.

75. Yadav Vikas, Bethard Steven, Surdeanu Mihai. Unsupervised Alignment-based Iterative Evidence Retrieval for Multi-hop Question Answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:4514–4525; 2020.

76. Yadav Vikas, Bethard Steven, Surdeanu Mihai. Quick and (not so) Dirty: Unsupervised Selection of Justification Sentences for Multi-hop 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):2578–2589; 2019.

77. Trivedi Harsh, Kwon Heeyoung, Khot Tushar, Sabharwal Ashish, Balasubramanian Niranjan. Repurposing Entailment for Multi-Hop Question Answering Tasks. 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):2948–2958; 2019.

78. Chen Qian, Zhu Xiaodan, Ling Zhen-Hua, Wei Si, Jiang Hui, Inkpen Diana. Enhanced LSTM for Natural Language Inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers):1657–1668; 2017.

79. Papineni Kishore, Roukos Salim, Ward Todd, Zhu Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics:311–318; 2002.

80. Lavie Alon, Agarwal Abhaya. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In: Proceedings of the second workshop on statistical machine translation:228–231; 2007.

81. Lin Chin-Yew. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out:74–81; 2004.

82. Koupaee Mahnaz, Wang William Yang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*. 2018;.

83. Jin Qiao, Dhingra Bhuwan, Liu Zhengping, Cohen William, Lu Xinghua. PubMedQA: A Dataset for Biomedical Research 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):2567–2577; 2019.

84. Savery Max, Abacha Asma Ben, Gayen Soumya, Demner-Fushman Dina. Question-driven summarization of answers to consumer health questions. *Scientific Data.* 2020;7(1):1–9.

85. Hasselqvist Johan, Helmertz Niklas. Query-Based Abstractive Summarization Using Neural Networks. Master's thesis2017.

86. Ni Ansong, Azerbayev Zhangir, Mutuma Mutethia, et al. SummerTime: Text Summarization Toolkit for Non-experts. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations:329–338; 2021.

87. Pennington Jeffrey, Socher Richard, Manning Christopher D. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP):1532–1543; 2014.

AUTHOR BIOGRAPHY

Mahsa Abazari Kia received the M.S. degree in software engineering from the University of Isfahan, Isfahan, Iran, in 2018, and the B.S. degree in computer science from the University of Isfahan, Isfahan, Iran, in 2014. She is currently pursuing the Ph.D. degree in computer science with the University of Essex funded by BT (formerly British Telecom) and the University of Essex exploring novel methods in multi-document text summarization. Her research interests include text mining, natural language processing, and computer vision.

Aygul Garifullina received the BEng and MEng degrees in telecommunications from Kazan State Technical University, Russia, in 2007 and 2009 respectively, and the MSc degree in communications and signal processing from Newcastle University, UK, in 2009. She is currently a Research Manager, Applied Research, BT, UK. Her current research interests include text analytics, natural language processing and machine learning applied to desk-based operations improvement.

Mathias Kern received his MSc and Ph.D. in Computer Science from the University of Essex, UK, in 1998 and 2006, respectively. He is currently Senior Research Manager for sustainable resource management and optimization in the Applied Research team of BT, UK. He is an experienced industrial researcher and a strong advocate for both Artificial Intelligence and Operational Research technologies and the way they interact and can be applied to real-life problems, with a particular focus on sustainable operations to help BT achieve its net-zero ambitions. He is an active member of the Operational Research and the British Computer Society and represents BT on both the OR Society's Analytics Development Group (ADG) and the Heads of OR and Analytics Forum (HORAF).

Jon Chamberlain is a Senior Lecturer working in the School of Computer Science and Electronic Engineering at the University of Essex. He works on interdisciplinary research in the fields of human-computer interaction (HCI), natural language processing, and collective intelligence. His research interests include the identification of reliable information from a range of data sources, applying methods such as information extraction and aggregation on different input signals. In addition to his academic background, he has a track record of developing and supporting research-focused Web applications with industry partners such as BT, SignalAI, and Natural England.

Shoaib Jameel received the Ph.D. degree from The Chinese University of Hong Kong. He is currently a Lecturer in computer science and artificial intelligence at the School of Computer Science and Electronic Engineering, University of Essex, U.K. He works with various technology startups in the U.K., spearheading their technical sphere, where his research outputs are directly applied to their production systems. His works have appeared in various prestigious conferences and journals, such as SIGIR, AAAI, ACL, IJCAI, and TOIS. His research interests include text mining, natural language processing, and computer vision. He is a fellow of the Higher Education Academy.