IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received March 22, 2022, accepted April 19, 2022, date of publication April 25, 2022, date of current version May 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3170466

Adaptable Closed-Domain Question Answering Using Contextualized CNN-Attention Models and Question Expansion

MAHSA ABAZARI KIA[1], AYGUL GARIFULLINA[2], (Member, IEEE), MATHIAS KERN[2], JON CHAMBERLAIN[1], AND SHOAIB JAMEEL[3]
[1]School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.
[2]BT, Adastral Park, Ipswich IP5 3RE, U.K.
[3]Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Corresponding author: Mahsa Abazari Kia (ma19194@essex.ac.uk)

This work was supported in part by the Computer Science and Electronic Engineering Department, University of Essex; and in part by BT (formerly British Telecom) Group.

ABSTRACT In closed-domain Question Answering (QA), the goal is to retrieve answers to questions within a specific domain. The main challenge of closed-domain QA is to develop a model that only requires small datasets for training since large-scale corpora may not be available. One approach is a flexible QA model that can adapt to different closed domains and train on their corpora. In this paper, we present a novel versatile reading comprehension style approach for closed-domain QA (called CA-AcdQA). The approach is based on pre-trained contextualized language models, Convolutional Neural Network (CNN), and a self-attention mechanism. The model captures the relevance between the question and context sentences at different levels of granularity by exploring the dependencies between the features extracted by the CNN. Moreover, we include candidate answer identification and question expansion techniques for context reduction and rewriting ambiguous questions. The model can be tuned to different domains with a small training dataset for sentence-level QA. The approach is tested on four publicly-available closed-domain QA datasets: Tesla (person), California (region), EU-law (system), and COVID-QA (biomedical) against nine other QA approaches. Results show that the ALBERT model variant outperforms all approaches on all datasets with a significant increase in Exact Match and F1 score. Furthermore, for the Covid-19 QA in which the text is complicated and specialized, the model is improved considerably with additional biomedical training resources (an F1 increase of 15.9 over the next highest baseline).

INDEX TERMS Closed-domain question answering, convolutional neural network, question expansion, self-attention.

I. INTRODUCTION

In automated Question Answering (QA), the goal is to retrieve answer(s) to a particular question expressed as a natural language text [1]. In closed-domain QA, the focus is on a particular domain of interest where the goal is to retrieve answers to questions within that domain. Machine reading comprehension (MRC) is the core task for textual QA, which aims to infer the answer for a question given the related context [2]. The answers could be sentences or paragraphs, or even n-grams. In practice, sentences are a good size to present a user with a detailed answer. For instance, given the question "Why is the Pfizer vaccine better than Sinovac?", one would expect the answer in one or two sentences rather than a single phrase. The task is more challenging compared to others in information retrieval (IR) [3], where the goal is to retrieve a ranked list of relevant documents.

The focus of this paper is closed-domain sentence-level QA [4]–[7]. This is an important and challenging field to study because many problems could be addressed by building domain-specific QA systems. For example, technology companies building systems for their call agents to answer user queries would benefit from a system that uses their internal call records so their call agents could efficiently get an answer to the questions of their clients. In a further example, students studying a particular subject would benefit

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

from closed-domain QA systems to help them answer questions surrounding their syllabus, rather than using a general open-domain QA system that might retrieve irrelevant answers due to the diversity of topics covered.

Developing methods to improve closed-domain QA is a crucial problem to address so that we can build systems that answer domain-specific user questions effectively. It is challenging because there are a variety of domains, each with its vocabulary, language syntax, and semantics. Ideally the same computational model would be applied in different domains with minimal human supervision to avoid needing tailor-made models for every domain, which would be time-consuming and expensive. In automated QA, there has been significant progress, concentrated largely on the open-domain QA systems. There are systems in both open and closed-domain QA that have used popular pre-trained neural contextual language encoders such Bidirectional Encoder Representations from Transformers (BERT) [8] and other variants [9], [10]. The language models have achieved near-human, or even better performance, on popular open-domain QA tasks such as SQuAD [11]. Despite this progress in open-domain QA, existing models for closed-domain QA [4]–[7], [12] are comparatively less effective and open-domain QA models do not perform as expected for domain-specific questions. Our goal in this paper is to develop a closed-domain QA system that can be easily adapted to different domains with only a small training data set.

We propose an adaptable closed-domain MRC-style QA system based on a Convolutional Neural Network (CNN) and self-attention mechanism, with several characteristics that tackle contextual understanding for closed-domain QA. To enable the model to focus on question-relevant sentences, we apply an unsupervised filtering technique to remove those sentences which do not contain an answer for the question. The model also attempts to rewrite some question (determined by a tuned parameter) to make them less ambiguous. Closed-domain QA does not typically have large-scale datasets that could help develop a statistical model and, as a result, many strong open-domain QA models will struggle in closed domains. Applying statistical learning models on small datasets also introduces the problem of reliable generalization thus, we divide the fine-tuning process into two steps: 1) transfer to the task; and 2) adapt to the target domain. The two-step fine-tuning process addresses the data scarcity problem for closed-domain QA. The first fine-tuning step only needs to be done once, but the second step is required each time we adapt the model to a new domain.

To the best of our knowledge, there is no existing flexible system that can effectively adapt to different closed QA domains. Our contributions are as follows:

1) A novel hierarchical CNN attention network for reading comprehension style QA, which aims to answer questions at sentence-level for a given context in a specific domain. The CNN-attention model extracts local and mutual interactions among different words and phrase-level correlations to comprehend context sentences and questions. A candidate answer identifier module and a question expansion module for selecting question-relevant sentences and question rewriting, respectively, were also incorporated.
2) The approach is compared against state-of-the-art comparative methods for closed-domains QA and MRC tasks, and also models for open-domain QA. The results on different domains show that open-domain QA models are not very effective when there is sparse data and cases when questions are domain-dependent.
3) An ablation study highlighting the utility of the candidate answer identification and question expansion techniques that could be used to augment other approaches.

II. RELATED WORK

A. OPEN-DOMAIN QA

Lin *et al.* [13] developed a distantly supervised open-domain QA model that utilises an information retrieval-based paragraph selector to filter out noisy paragraphs and a paragraph reader to extract the correct answer using a multi-layer long short-term memory network. Yang *et al.* [14] demonstrated an end-to-end question answering system that integrates a BERT-based reader with the open-source Anserini information retrieval (IR) toolkit to identify answers from a large corpus of Wikipedia articles in an end-to-end fashion. Karpukhin *et al.* [15] focused on establishing the optimal training procedure utilising a sparse set of question and passage pairs. They designed retrieval solely through dense representations, with embeddings learnt from a modest number of questions and passages using a simple dual-encoder system. Seo *et al.* [16] introduced Dense-Sparse Phrase Index (DENSPI), an indexable query-agnostic phrase representation model for real-time open-domain QA on SQuAD. In their model, phrase representation combines dense and sparse vectors based on BERT and term-frequency-based encoding, respectively. Qu *et al.* [17] proposed an open-retrieval conversational QA (ORConvQA) containing a retriever, reranker, and a reader that are all based on fine-tuned BERT and ALBERT based encoders and decoders. They evaluated their model on the OR-QuAC dataset they created for conversational QA.

B. CLOSED-DOMAIN QA

Lende and Raghuwanshi [6] proposed a system for closed-domain QA for user queries related to education. An index term dictionary was created for the keywords extracted from a corpus created for the education domain. To obtain the relevant answer, they apply Part-of-speech (POS) tagging to all the filtered documents to find the suitable answer, which contains the same sense as the query. Sarkar *et al.* [7] developed a knowledge-based QA system, which only understands predefined insurance-related queries. In the first step, the Apache OpenNLP tool is used to detect the query's subject-to-predicate triplets, and then relevant content was retrieved and ranked using matching criteria (query sentence similarity, sentence length, relative word importance, etc.). Badugu and Manivannan [5] created a closed-domain question answering framework for "Hyderabad Tourism" based on rule-based classification and similarity measures. The corpus is preprocessed, divided into sentences, and then grouped into various inquiry types such as *What*, *Where*, *Who*, and *When*. Sentence retrieval is conducted, based on the question category, and their vectors are generated based on the term frequency and the inverse document frequency of the term. The Jaccard similarity score determines the final answer for each question. A BERT-based clinical question answering system was proposed by Rawat *et al.* [4], using fine-tuned BERT on medical corpora. Entity-level clinical concepts were integrated into the BERT architecture using the Enhanced Language Representation with Informative Entities (ERNIE) framework. ERNIE extracts contextualized token embeddings using BERT and generates entity embeddings using a multi-head attention model. Godavarthi and Sowjanya [18] built a closed-domain QA system that answers queries from the COVID-19 open research data set (CORD-19). They fine-tuned a BERT model for self-supervised learning of language representations (ALBERT) [19] for retrieving all COVID relevant information to the query. Cai *et al.* [20] proposed an integrated framework for answering Chinese questions in restricted domains by modeling the question pair, comparing the input question to the existing question, and then identifying the answer output.

C. READING COMPREHENSION APPROACHES

Reading approaches can be classified into two categories based on whether the retrieved documents are processed independently or jointly for answer extraction. This subsection summarizes recent reading comprehension approaches (readers) in different QA models. With the use of BERT Reader, Dense Passage Retrieval [15] estimates the likelihood that a passage contains the answer and the probability that the token is the beginning and end of an answer span. It then selects the most probable answer based on what it calculates. Readers are often developed as graph-based systems to extract answer spans from passages [21], [22]. For example, in Graph Reader [22], the graph is used as input, and Graph Convolution Networks [23] are primarily used to learn the passage representation before pulling the answers from the most probable span. In DrQA [24], various features, such as POS, named entities (NE), and term frequencies (TF), are extracted from the context. The multilayer Bi-LSTM then predicts the span of the answer based upon the inputs, the question, and the paragraphs. As part of this process, argmax is applied across all answer spans to get a final average of answer scores across paragraphs using an un-normalized exponential function. BERTserini [14] provides a reader that works on BERT by removing the softmax layer, which allows for comparison and aggregation across different paragraphs. A Shared Normalization mechanism modifies the objective function and normalizes the start and end scores across all paragraphs to achieve consistent performance gains [13]. This mechanism eliminates the problem of unnormalized scores (e.g., exponential scores or logit scores) for all answer spans.

D. QUERY EXPANSION

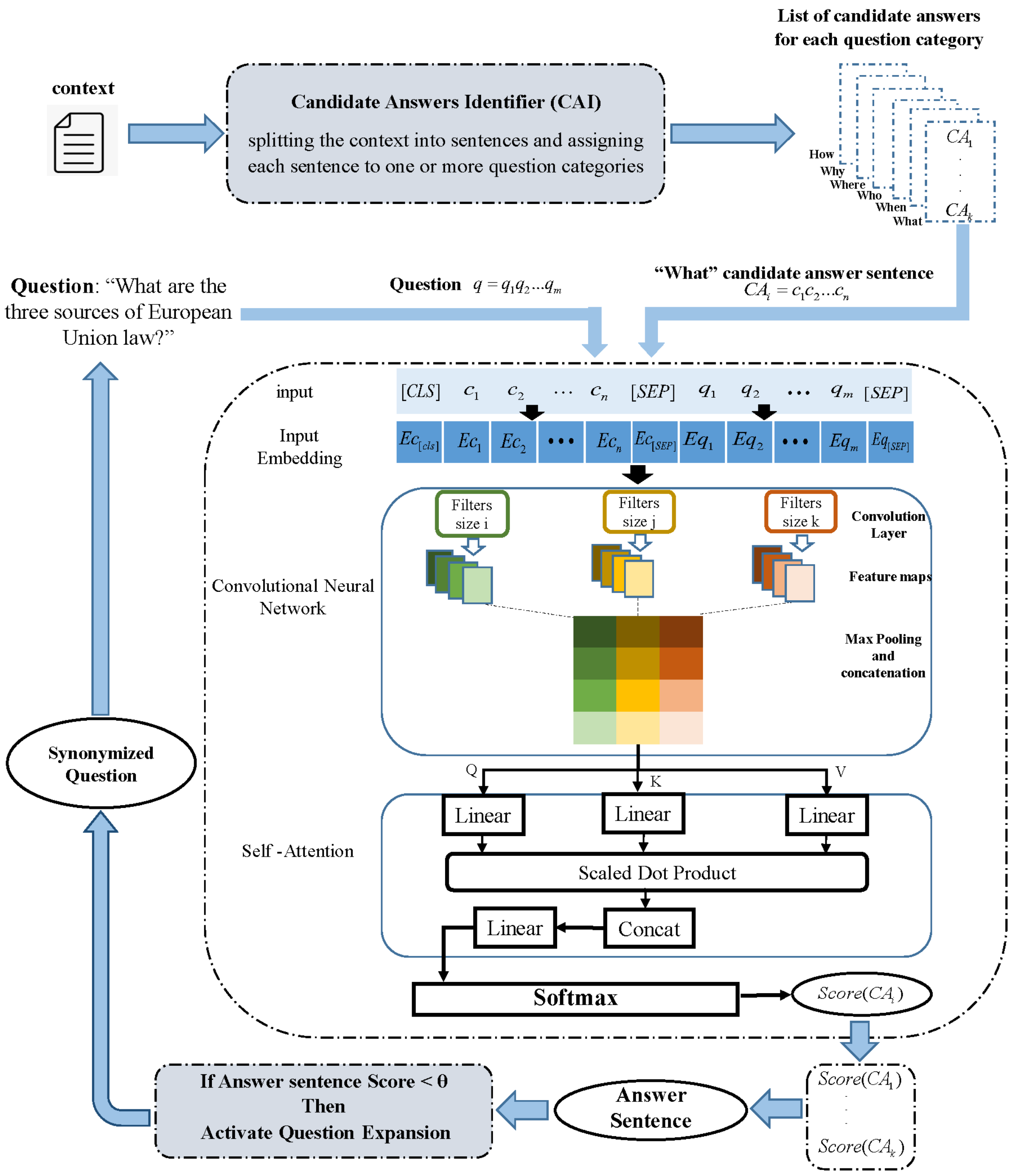
This subsection explains recent question expansion (reformulation) approaches proposed for QA systems. According to GOLDEN Retriever [25], the query reformulation task can be recast as an MRC task because they both take a question and some context documents as inputs and aim to generate natural language strings as outputs. The query expansion module in GAR [26] is built using a pre-trained Seq2Seq model BART [27] to take the original query as input and generate new queries. The model is trained with various generation targets: the answer, the sentence containing the answer, and the passage title. Some other works generate dense representations to be used for searching in a latent space. For example, Multi-step Reasoner [28] employed a Gated Recurrent Unit (GRU) [29], taking token-level hidden representations from MRC and the question as input to generate a new query vector. The new query vector is then trained using Reinforcement Learning (RL) by comparing the extracted answer to the ground-truth. Xiong *et al.* [30] uses a pre-trained masked language model (such as RoBERTa [31]) as its encoder, which concatenates all previous passages and the question representation to encode a dense query.

III. ADAPTABLE CLOSED-DOMAIN QA MODEL

In this section, we describe our novel reading comprehension model for sentence-level closed-domain QA that can be tuned with a small training dataset for various domains. We have introduced a candidate answer identifier (CAI) module based on syntactic and linguistic rules to reduce the context to the sentences that could contain the answer for the given question (candidate answer sentences). We designed a neural network based on CNN and self-attention mechanism to analyze and score the candidate answer sentences. The novelty lies in obtaining different levels of contextual understanding of context sentences and the question by extracting important semantic features and their correlations. The CNN-attention layer assigns a relevance score for each candidate answer sentence selected by the CAI. Also, we have introduced a question expansion module (QE) for rewriting ambiguous questions shown in Fig. 3, which in spirit, is close to the query expansion technique in Information Retrieval. The key advantage of this module is that it rewrites the question and produces synonym versions to help the system select the answer sentence with more confidence. The overall framework of our model is shown in Fig. 1.

A. CANDIDATE ANSWERS IDENTIFIER (CAI)

Unlike previous approaches, we filter out irrelevant content from context P to help improve our results. We analyze the linguistic features for each sentence to determine its capability for answering different question categories



List of candidate answers for each question category
context
Candidate Answers Identifier (CAI)
splitting the context into sentences and assigning each sentence to one or more question categories
How
Why
Where
Who
When
What
CA_1
CA_k
Question: "What are the three sources of European Union law?"
Question $q = q_1 q_2 \dots q_m$
"What" candidate answer sentence $CA_i = c_1 c_2 \dots c_n$
input
$[CLS]$ c_1 c_2 … c_n $[SEP]$ q_1 q_2 … q_m $[SEP]$
Input Embedding
$Ec_{[cls]}$ Ec_1 Ec_2 ••• Ec_n $Ec_{[SEP]}$ Eq_1 Eq_2 ••• Eq_m $Eq_{[SEP]}$
Filters size i
Filters size j
Filters size k
Convolution Layer
Feature maps
Max Pooling and concatenation
Convolutional Neural Network
Synonymized Question
Q
K
V
Linear
Linear
Linear
Self -Attention
Scaled Dot Product
Linear
Concat
Softmax
$Score(CA_i)$
$Score(CA_1)$
$Score(CA_k)$
If Answer sentence Score < θ Then Activate Question Expansion
Answer Sentence

FIGURE 1. The overall framework of our model CA-AcdQA for an example "What" question.

(*When*, *Where*, *Who*, *What*, *Why*, *How*). We developed a strategy to classify the context sentences into question categories to facilitate the answer selection. To this end, we use a popular tool called Giveme5W1H [32], an open-source system that uses syntactic rules to automatically extract the relevant phrases from English news articles for answering the 5W1H questions. The advantage of this tool is that it can be customized towards one's needs. Since our main goal is identifying candidate answers for each question category, we have customized different components in Giveme5W1H functions. We have designed new methods and rules to improve and adapt the Giveme5W1H for candidate answer selection since the Giveme5W1H does not cover all the syntactic rules for "Why", "What", and "Where". Also, we have used a new parser function for finding all types of date-time named entities (NEs) for "When". Additional methods are added to Giveme5W1H to support all types of "How" questions such as "How many", "How much" and "How". We perform six independent identification functions to retrieve the candidate answers for the six (5W1H) categories. The candidate answer identifier module uses the Giveme5W1H preprocessing steps, gets the context as input, and splits it into sentences to process them separately. After checking all the rules and methods for each sentence, it will be added to correspondent categories,

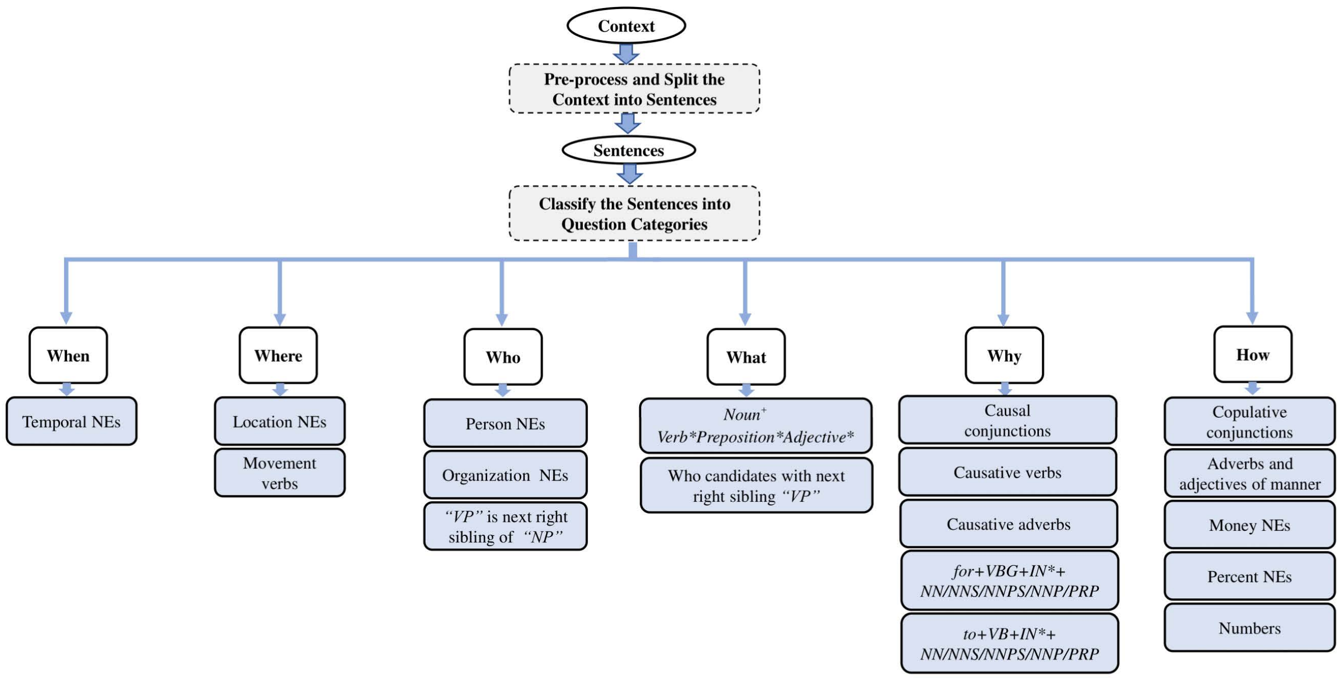

Context
Pre-process and Split the Context into Sentences
Sentences
Classify the Sentences into Question Categories
When
Temporal NEs
Where
Location NEs
Movement verbs
Who
Person NEs
Organization NEs
"VP" is next right sibling of "NP"
What
Noun+ Verb*Preposition*Adjective*
Who candidates with next right sibling "VP"
Why
Causal conjunctions
Causative verbs
Causative adverbs
for+VBG+IN*+ NN/NNS/NNPS/NNP/PRP
to+VB+IN*+ NN/NNS/NNPS/NNP/PRP
How
Copulative conjunctions
Adverbs and adjectives of manner
Money NEs
Percent NEs
Numbers

FIGURE 2. Processing the context and identifying the candidate answers for each question category based on linguistic and syntactic patterns and features.

and in the end, a list of candidate answers for each question category will be prepared. More details on different rules and how they have been incorporated into our candidate answer identifier module are mentioned below and depicted in Fig. 2.

- When: For detecting all types of temporal NEs including all formats of DateTime, duration, etc we have added the dateparser[1] python package, as well as using SUTime [33] which is used in Giveme5W1H.
- Where: Giveme5w1H only looks for sentences containing tokens classified as NEs of the type Location. We have added a new method searching for movement verbs like "go", "move", "run", "jump", "bolt", and others, following by a preposition ("to", "toward", etc.) to point to a location.
- Who: In Giveme5W1H, the sentences that have the subject are considered as "who" candidate answers. The first noun phrase (NP) that is a direct child to the sentence in the parse tree and has a verb phrase VP) as its next right sibling is the sentence subject. We also considered the sentences containing Person or Organization NEs as "who" candidate answers which are missed in the original Giveme5w1H.
- What: In Giveme5W1H, the "who" candidates that a VP is the next right sibling in their parse tree are considered as "what" candidates. We have extended this function because the original function does not reliably work on many "what" candidate answers. We added an extra function to select sentences as the candidate answers for "what"; however, the order of tags is not important. The pattern that we look for is $(Noun^{+}Verb^{*}Preposition^{*}Adjective^{*})$.
- Why: In Giveme5W1H, sentences containing causal conjunctions ("due to", "result of", "because" and "effect of"), causative adverbs ("therefore", "hence", and "thus"), causative verbs ("activate", "implicate", "make to", etc.) are considered as "why" candidates. We added two syntactical rules for covering all "why" candidate answers. The sentences containing the following sequence(s) are "why" candidate answers. The patterns that we look for are:[2]
$(``to + VB + IN^{*} + NN/NNS/NNPS/NNP/PRP")$
$(``for + VBG + IN^{*} + NN/NNS/NNPS/NNP/PRP")$.
- How: Giveme5W1H proposed a combined method consisting of two subtasks, one analyzing copulative conjunctions, the other looking for adjectives and adverbs of manner for the "How" category. We added an extra method to search for Money NEs, Percent NEs, or numbers as the candidate answers for "How many" and "How much".

The candidate answer sentences list for questions that do not belong to the 5W1H categories contains all the context sentences.

[1]https://github.com/scrapinghub/dateparser

[2]VB, VBG, NN, NNS, NNP, NNPS, PRP, and IN stand for base form verb, present participle verb, singular noun, plural noun, singular proper noun, plural proper noun, personal pronoun, and Preposition or subordinating conjunction respectively.

B. CNN-ATTENTION BASED ANSWER SELECTOR

Given the question q, represented as a sentence, there are K possible candidate answers $CA_1, CA_2, \ldots, CA_k$ which are present in the accompanying context P associated with the q. Question q with m tokens ($q = q_1, q_2, \ldots, q_m$) and candidate answer sentence CA_i with n tokens ($CA_i = c_1, c_2, \ldots, c_n$) are combined together into a single sequence, separated by a special token $[SEP]$ as the input of the CNN attention layer. The output of BERT is taken only for the first token $[CLS]$, which is used as the aggregate representation of the sequence. We derive the semantic representation of q and CA_i using a pre-trained contextual language model such as BERT or ALBERT for the embedding layer. The advantage is that we derive high-quality representations, which cannot be obtained using methods such as static word embeddings [34], [35]. Our goal is to obtain a reliable or most plausible answer CA_j to the question q in P. BERT uses a multi-layer bidirectional transformer [36] network to encode contextualized language representations. Similar to BERT, the ALBERT model introduces two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT. To calculate the scores for candidate answer sentences, we fine-tune the BERT and ALBERT pre-trained model with untrained layers of CNN, pooling, and attention. The CNN and self-attention mechanism focus the model on the most important features and their correlations when constructing the question and sentence representation.

1) CONVOLUTIONAL NEURAL NETWORK

The CNN extracts salient n-gram features from the input sentence to create an informative latent semantic representation of the sentence for downstream tasks [37]. For each sentence, let $e_i \in R^d$ represent the word embedding for the i^{th} word in the sentence, where d is the dimension of the word embedding, and the given sentence has n words. Convolution is then performed on this input embedding layer. It produces a new feature by applying a filter $K \in R^{hd}$ of size h on a window of h words.

For example, a feature c_i is generated using the window of words $e_{i:i+h-1}$ by (1).

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

Here, f is a non-linear activation function, for example, the hyperbolic tangent, and $b \in R$ is the bias term. The filter (also called kernel) K is applied to all possible windows (slide over the entire sentence embedding matrix) using the same weights to create the feature map. We divide the sentence of length n into $\{e_{1:h}, e_{2:h+1}, \ldots, e_{i:i+h-1}, \ldots, e_{n-h+1:n}\}$ and perform the filter on each component. The feature map obtained by filter is shown in (2).

$$c = [c_1, c_2, \ldots, c_i, \ldots, c_{n-h+1}] \tag{2}$$

A convolution layer is usually followed by a pooling strategy on each filter to provide a fixed-length output and reduce the output's dimension while retaining the most salient features. In this paper, the maximum pooling method on each feature map is applied, which gives us low dimensions dominant features, as shown in (3).

$$\hat{c} = max\{c\} \tag{3}$$

The $\hat{c}$ is obtained by one convolution filter along with maximum pooling layer, and a feature sequence obtained with t convolution filters is shown in (4).

$$\hat{C} = [\hat{c_1}, \hat{c_2}, \ldots, \hat{c_t}] \tag{4}$$

In this stage, important n-gram features of the candidate answer sentence and question are extracted by CNN, and the generated feature vectors should be concatenated to form the new global feature vector matrix Y as the input to the self-attention layer.

2) SELF-ATTENTION LAYER

The self-attention mechanism primarily focuses on the internal dependence of input [38]. In our model, the self-attention layer calculates the semantic association between the extracted features from the question and candidate answer sentence to determine the candidate answer's relevance score. In each self-attention mechanism, there is a query matrix (Q), a key matrix (K) and a value matrix (V). The output of the CNN layer, matrix Y, is the the initial value of query matrix (Q), key matrix (K) and value matrix (V), as shown in (5).

$$Q = K = V = Y \tag{5}$$

Scaled Dot-product Attention (SDA) is the main concept of the self-attention mechanism. It first computes the similarity by solving the dot product of Q and K, then divides by $\sqrt{d_k}$ (d_k is the dimension of matrix K) to avoid the dot product result from being too large. The result is then normalized using the Softmax function before being multiplied by the matrix V to obtain the expression of attention. SDA operation is depicted in (6).

$$SDA(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

We perform average pooling on the output matrix of the self-attention layer to obtain the feature vector f for integrated CA_i and q. We input f through the fully connected layer to the final softmax layer. In the answer selection task, there are two classifier labels (similar = 1, dissimilar = 0). We modified the final layer to get the predicted $Score(CA_i)$ for the similar label, as shown in (7) and (8).

$$Score(CA_i) = P(C = 1|CA_i, q) \tag{7}$$

$$P(C|CA_i, q) = softmax(w_c f + b_c) \tag{8}$$

where w_c is the weight matrix, b_c is the bias and C is label. We rank all the candidate answer sentences based on the obtained scores, and the candidate answer sentence with the highest score is selected as the answer sentence for the question q. We prevent having more than one sentence with label 1 with this method.

C. QUESTION EXPANSION

Some questions are more ambiguous or convey less domain-related information than others [39]. Inspired by research in Information Retrieval, where query terms are expanded with relevant keywords from the vocabulary, we developed a strategy to use more appropriate terms if the question does not convey much information to our model. We introduce a parameter θ where $0 < \theta < 1$, which is automatically tuned from the data and helps us to assess whether question expansion is needed. If the selected answer sentence score is less than θ, the question expansion module generates question synonyms until a candidate answer achieves a score greater than θ. We have designed a lightweight hybrid question expansion based on contextualized embedding and lexical resources (WordNet) that replaces some question keywords with domain-related synonyms. We extract the question keywords by POS tagging the question and removing the symbols, stopwords, and NEs to keep the words most important to the question.

After selecting the keywords, expansion terms are extracted from WordNet considering the keyword's role in the question (for example, if the keyword is an adjective, adjective synonyms are selected accordingly). Thereafter ranking and filtering functions are applied to choose the most appropriate expansion terms for each keyword.

$$Question - Keywords = [K_1, K_2, \ldots, K_m] \tag{9}$$

$$\begin{aligned} expansion - list = {} & [(K_1 : et_1, \ldots, et_w) \\ & , \ldots, (K_m : et_1, \ldots, et_z)] \end{aligned} \tag{10}$$

The expansion terms that do not exist in the domain vocabulary are eliminated from the list, and the remaining ones are considered for calculating their relevance to the question. We further train the pre-trained BERT model using the domain-specific corpus to generate domain-specific embedding vectors for expansion terms and questions. We rank the expansion terms regarding their relatedness to the whole question, and those more semantically related to the question are retained. Therefore the semantic similarity between question and expansion terms embedding vectors is calculated.

After finalizing the expansion list, each expansion term is transformed to the appropriate form to get the same POS tag as the keyword (for example, if the keyword is plural Noun(NNS), its expansion term should be the same). Then, each keyword is replaced with one of the expansion terms to form a synonym version of the question that conveys the same context. The generated synonym versions have the same structure as the original question since only some keywords are replaced with their synonyms. As a result, there is no need to do grammar checking for the generated versions.

For example, *"What are main steps for mitigating the COVID -19 transmission during transport of suspected and confirmed patients?"* is a question from the COVID-QA dataset that needs expansion because its answer sentence score is less than θ. The first step is keyword selection, *{"main", "steps", "mitigating", "transmission", "transport", "suspected", "confirmed", "patients"}* are the question keywords and their expansion terms are extracted by using WordNet (synonyms with the same role as the keyword are selected for each keyword).

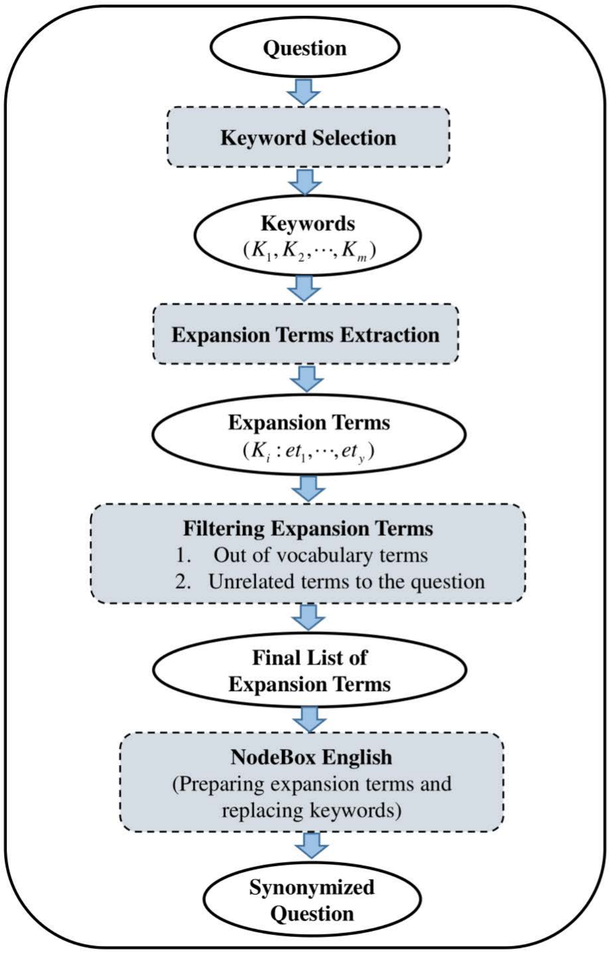

Question
Keyword Selection
Keywords
$(K_1, K_2, \cdots, K_m)$
Expansion Terms Extraction
Expansion Terms
$(K_i : et_1, \cdots, et_y)$
Filtering Expansion Terms
1. Out of vocabulary terms
2. Unrelated terms to the question
Final List of Expansion Terms
NodeBox English
(Preparing expansion terms and replacing keywords)
Synonymized Question

FIGURE 3. The overall steps of the question expansion (QE) module for rewriting ambiguous questions.

After the first step of domain vocabulary filtering, the list of question keywords and their domain-related expansion terms are *{ main(adj): major, primary, principal – transmission(noun): infection, contagion – transport(noun): transfer – confirmed(verb): corroborate, affirm, substantiate}*.

The second step of the filtering is to measure the expansion terms' semantic relevance to the question. We filtered the terms with lower relevance (lower semantic similarity) to the question for keywords with more than one synonym. The average of the semantic relevance to the question is calculated for all the keyword synonyms and those obtaining the semantic relevance more than the average value (α) will remain for the keyword. After this step, the final list of expansion terms with higher semantic relevance to the question remains *{main: major, primary - transmission: infection - transport: transfer - confirmed: corroborated, affirmed }*. We automatically transform the synonyms to their appropriate form to get the same POS tag as the keyword, "confirmed" has the "VBN: present participle" POS tag so its synonyms are converted to present participle form.

The synonym versions for the question are generated by replacing the keywords with their synonyms. One of the expanded versions of our example question is *"What are major steps for mitigating the COVID -19 infection during transfer of suspected and corroborated patients?"*.

Replacing the keywords (one adjective, two nouns, and one verb) with domain-relevant and question-related synonyms generates other versions of the question with the same meaning. We analyze the candidate answers for the generated synonym version of the question to find the answer sentence more accurately. The final selected answer sentence is *"HCWs who handle the transport of COVID-19 patients must consider the following principles: firstly, early recognition of the deteriorating patient; secondly, HCW safety; thirdly, bystander safety; fourthly, contingency plans for medical emergencies during transport; fifthly, post-transport decontamination."* with the score 0.72. If the scores for the selected answers by the synonymized versions are lower than θ, the answer sentence with the highest score (among all the selected answers) will be chosen. The question expansion module described in Algorithm 1 takes a question as input and generates the synonym version of the question in four steps: 1) keyword detection; 2) expansion terms (synonyms) extraction; 3) filtering inappropriate synonyms; and 4) preparing expansion terms and replacing keywords with their corresponded synonyms to generate various synonyms of the question.

Algorithm 1 Question Expansion

```
Input: Question (q)
Output: Synonym versions of Question
question-keywords=Removing stopwords, symbols and NEs(q)
for keyword in question-keywords do
    expansion-list.Add(find-synonyms(keyword,
    WordNet(keyword)))
end for
for term in expansion-list do
    if term is not in Vocabulary then
        expansion-list.Remove(term)
    end if
end for
for keyword in question-keywords do
    $\alpha$=AVG(CosineSimilarity(Emb(q),Emb(keyword.expansion-
    term(i))) {i in range Size(keyword.expansion-list)}
    {Emb(x) stands for Embedding vector for x}
    {With the $\alpha$, the synonyms which are more semantically
    related to the question will be selected for each keyword.}
    for term in keyword.expansion-list do
        if CosineSimilarity(Emb(q), Emb(term)) $<$ $\alpha$ then
            expansion-list.Remove(term)
        end if
    end for
end for
for expansion-term in expansion-list do
    {Preparing the expansion terms}
    expansion-term=NodeBox English(expansion-term,
    keyword.POStag)
end for
for expansion-term in expansion-list do
    synonym-version=Question.Replace(expansion-term,
    correspondent-keyword)
    Question-synonym-list.Add(synonym-version)
end for
return Question-synonym-list
```

IV. EXPERIMENTS

A. EXPERIMENTAL DATASET

We have used four closed-domain datasets to verify the performance of the proposed model. Three datasets were derived from popular SQuAD collection [11] due to the limited number of closed-domain QA datasets that are publicly available. The datasets are from three domains with different concepts and different sizes: Tesla (person); California (region); and European-Union-law (system) referred to as EU-law in our results. COVID-QA [40], a SQuAD style Question Answering dataset, was added as the fourth closed-domain dataset for our experiments. The datasets consist of Context-Answer-Question triples. The Tesla, California, EU-law, and COVID-QA consist of 565, 746, 315, and 2019 questions, respectively, along with annotated answers and context (see Table 1).

TABLE 1. The count of each question category for datasets.

Dataset	What	Where	When	Why	Who	How
Tesla	268	76	54	25	71	56
California	323	96	80	52	59	60
EU-law	136	17	23	23	39	18
COVID-QA	1335	53	53	80	22	272

B. DATA PRE-PROCESSING AND EXPERIMENTAL SETTINGS

We used Stanford CoreNLP [41] for sentence splitting, tokenization, full parsing, POS-tagging, preprocessing, and preparing the context in the CAI module for candidate answers selection. We used two-step training for the contextualized CNN-attention answer selector model: 1) transfer to the task; and 2) adaptation to the target domain. Performing single fine-tuning requires a large dataset for the target domain, which is impractical due to the difficulty and cost of collecting training data specific to that domain. Thus, the first step transfers the model to the target task, and the second step can adapt the model to the target domain with a small training dataset. We have utilized the Natural Questions (NQ) dataset [42] consisting of 300,000 naturally occurring questions, along with human-annotated answers from Wikipedia pages, to be used for the first step of training. This dataset provides a whole Wikipedia page for each question which is significantly longer compared to MRC datasets (e.g., SQuAD). Following Liu *et al.* [43], we generate multiple document spans by splitting the Wikipedia page using a sliding window with the size and stride 512 and 192 tokens respectively to generate the negative (i.e., no answer) and positive

TABLE 2. Optimal hyperparameters for Tesla, California, EU-law, and COVID-QA datasets. The search spaces are (0, 1), {2, 3, 4, 5}, {10, 20, 30} {1e − 7, 2e − 7, 1e − 8, 2e − 8, 5e − 8}, {4, 8, 16} for θ, filter size, number of filters, learning rate, and batch size respectively.

Hyperparameters	Tesla	California	EU-law	COVID-QA
θ	0.76	0.74	0.71	0.67
learning rate	1e-8	2e-8	1e-8	5e-8
filter size	2,3,4	2,3,4	2,3,4	2,3,4
filter number	20	20	20	30
batch size	8	8	4	16

(i.e., has answers) spans. Then, we only preserve the positive spans (the span containing the annotated short answer) as the context, and the negative ones were discarded. For both the first and second steps of training, the question sentence pairs were generated by CAI. After generating the candidate answer sentences for question categories with CAI, the candidate answer sentence and question pairs were generated for training the CNN attention-based answer selector. The candidate answer sentence which contains the annotated answer gets the label 1, and other candidate sentences get label 0. The first fine-tuning step is done only once, and the second step is performed each time we adapt the model to a new domain. We used the pre-trained BERT base and ALBERT base model for token embeddings, consisting of 12 Transformer blocks with 12 self-attention heads and the hidden size of 768. There is no analytical formula to calculate an appropriate value of the hyperparameters to obtain the optimal model parameter. Therefore, we used tools to automatically tune the model hyperparameters. We performed hyperparameter optimization using Ray Tune Python library[3] with Hyperopt algorithm [44]. Filter size, number of filters, learning rate, batch size, and theta (QE threshold) hyperparameters were optimized for each domain shown in Table 2. The search spaces are (0,1), {2, 3, 4, 5}, {10, 20, 30}, {1e-7, 2e-7, 1e-8, 2e-8, 5e-8}, {4, 8, 16} for θ, filter size, number of filters, learning rate, and batch size respectively. The optimal combination of hyperparameters values that maximize the model performance is discovered by the Hyperopt algorithm for each time tuning the model for a new domain. The Hyperopt algorithm utilizes a form of Bayesian optimization and requires the search space, the loss function, the optimization algorithm, and a database for recording hyperparameter tuning history (score, configuration). We set the maximum sequence length for BERT and ALBERT to 128 tokens. We utilized the Adam optimization algorithm [45] for the parameter update. We used the cross entropy loss function to calculate the loss. The optimal values for filter size, number of filters, learning rate, and batch size for the first step of training are calculated as follows: {2, 3, 4}, 100, 2e-5, 64. We applied early stopping on the development set for both training stages on the loss value. We set the max number of epochs to 9 and 3 for transfer and adapt steps, respectively. For the QE module, we used domain-specific corpora (concatenation of contexts for one domain) for tuning the pre-trained BERT for generating domain-specific embeddings. We automatically prepared the domain-specific corpus for ''masked Language Model'' and ''next sentence prediction'' to generate the data for pre-training on each domain. We utilized the NodeBox English library, which has been succeeded by the Pattern Python library,[4] for analyzing the keyword's role and expansion term transformation.

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

C. EVALUATION METRICS

We adopt two metrics including Exact Match (EM) and F1 scores to evaluate our model. The EM score determines the percentage of predictions that perfectly match the ground truth answer, and the F1 score demonstrates the average overlap between the prediction and the ground truth answer.

D. COMPARATIVE METHODS

To demonstrate the effectiveness of our proposed model, we compare against several other comparative approaches:

- KPOS-QA [6] is a closed-domain QA system (their dataset is not publicly available). We have simulated their approach for sentence-level QA regarding the details provided in their paper (ranking and selecting the answer based on extracted keywords and POS tags for query and context).
- R-TFIDF [5] is another closed-domain QA system (their dataset is also not publicly available). We simulated their approach for sentence-level QA regarding the details provided in their paper (a rule-based sentence classification and measuring cosine similarity on TF-IDF vectors for question and sentences).
- AttReader [46] presented BiLSTM networks based on an attention mechanism and the GLoVe language model for reading comprehension in QA.
- QANET [47], is an MRC model for open-domain QA based on convolutions, global self-attention, and the GLoVe language model.
- cdQA is an end-to-end closed domain QA system built on top of the pre-trained BERT.[5]
- Retro-reader [48] is an ''open-domain'' MRC model and ranks 5th in the SQuAD2.0 leaderboard.[6] An approach with two reading modules (sketchy reading module and intensive reading module) is proposed to find answer span and detect unanswerable questions. In the intensive reading module, two question-aware matching mechanisms based on the transformer and multi-head attention are introduced for predicting the answer.
- ZCovid-QA [49], employed RoBERTa fine-tuned on the SQuAD and QuAC datasets for zero-shot evaluation on the COVID-QA dataset for Covid-19 QA.

[4]https://github.com/clips/pattern
[5]https://github.com/cdqa-suite/cdQA
[6]We did not find openly available source codes of other top-ranking models even after contacting their authors. As a result, we compare our method with the model whose code we could obtain.

- EtoE-Covid-QA [50] fine-tuned RoBERTa-large on SQuAD2.0, NQ, and proposed both language modeling on the CORD-19 collection and example generation model for the MRC training for Covid-19 QA.
- OCovid-QA [51] utilized a variant of BioBERT fine-tuned on the SQuAD2.0 and COVID-QA datasets for Covid-19 QA.

V. RESULTS AND DISCUSSION

We present the results obtained by our model and others on the development set in Table 3. We present results for two variants of our model (CA-AcdQA): 1) pre-trained BERT; and 2) pre-trained ALBERT. For AttReader, QANET, cdQA, Retro-Albert, we used their public code to apply their model to the datasets and generate results for closed-domain sentence-level QA. We used the same pre-trained language models as reported in their respective papers and fine-tuned the models with two stages of training as mentioned in IV-B. ZCovid-QA, EtoE-Covid-QA, and OCovid-QA baselines are only designed for Covid-19 QA and the results for these models are reported in their respective papers. We categorized the comparative models into two groups: 1) based on conventional language models (KPOS-QA, R-TFIDF, AttReader, QANET); and 2) contextualized language models (cdQA, RetroReader, ZCovid-QA, EtoE-Covid-QA, OCovid-QA). We observe from our results that KPOS-QA, which is based on context and question keyword extraction using POS tags, achieves the worst results. Hence, we learn that there is a strong need for high-quality vectors representing context and question. In R-TDIDF, we notice an improvement of 3%-11% of the F1 score, obtained by applying traditional TF-IDF vectors and sentence classification. The QANET and AttReader outperformed the KPOS and R-TFIDF, whereas the pre-trained GLoVE language model encodes context and question. The QANET outperformed AttReader because it's not relying on the recurrent structure, unlike the AttReader, which is based on BiLSTM. cdQA outperformed QANET and AttReader and improved the EM due to its reader architecture based on BERT. RetroReader outperformed the other baseline methods for all datasets since it employed a pre-trained transformer-based language model and attention mechanism for reading comprehension. Evaluating RetroReader for closed-domain reading comprehension shows its performance has degraded slightly (its performance in open-domain QA is 91.3 for F1 score and 88.8 for EM). Our model outperforms all baseline models for all datasets because we explore the association between the extracted features from the question and candidate answer sentences by applying CNN and self-attention on the joint representation of question and candidate answer sentences. Also, the CAI and QE module's effect on selecting appropriate sentences from context and rewriting the vague questions should not be disregarded. The performance of all baseline methods is worse on the COVID-QA dataset since Biomedical QA (BQA) is more challenging than other domains, and more reasoning is needed for the question and biomedical text compared to other domains. Another challenge is clinical term ambiguity due to the variation of clinical terminology and the frequent use of abbreviations and esoteric medical terminology. BQA evaluation is also challenging because most evaluation metrics do not consider the rich biomedical synonym relationships. Since biomedicine is a highly specialized domain, understanding complex biomedical knowledge is required, and using contextualized language models pre-trained on open-domain corpora is inefficient. We evaluated our approach utilizing pre-trained BERT on the biomedical domain as shown in Table 3. SciBERT [52] is trained on a large corpus of scientific text, including text from the biomedical domain, and BioBERT [53] is the first domain-specific BERT-based model pre-trained on biomedical corpora. DeepSet[7] has made available a BERT-base model pre-trained on CORD-19 [54], and it is evident that pre-training BERT with CORD-19 corpus improves our model performance since this model is certainly more "in-domain" than BioBERT-base or SciBERT-base for COVID-19 QA. OCovid-QA outperformed EtoE-Covid-QA and ZCovid-QA since it is based on BioBERT, which is more appropriate for Covid-19 QA than RoBERTa used in EtoE-Covid-QA and ZCovid-QA.

TABLE 3. Performance comparison of CA-AcdQA models against other baselines.

	Tesla	California	EU-law	COVID-QA
Model	EM / F	EM / F	EM / F	EM / F
KPOS-QA	53.2 / 61.1	52.0 / 60.3	48.4 / 57.2	0 / 9.1
R-TFIDF	63.1 / 70.8	63.4 / 70.9	60.1 / 68.8	0 / 12.3
AttReader	71.3 / 79.5	70.8 / 79.8	68.9 / 78.7	11.2 / 41.4
QANET	75.2 / 83.3	75.3 / 83.1	73.8 / 82.2	12.4 / 43.6
cdQA	80.0 / 84.2	80.3 / 84.8	78.2 / 83.0	33.5 / 65.9
RetroReader	87.4 / 90.1	86.9 / 90.3	86.5 / 89.9	52.4 / 74.0
ZCovid-QA [49]				25.9 / 59.5
EtoE-Covid-QA [50]				38.6 / 62.8
OCovid-QA [51]				39.1 / 72.0
CA-AcdQA (BERT)	**89.8 / 93.2**	**89.5 / 92.8**	**88.0 / 92.2**	**55.6 / 76.4**
CA-AcdQA (ALBERT)	**92.6 / 95.5**	**92.3 / 95.3**	**91.2 / 95.4**	**57.5 / 78.8**
CA-AcdQA (SciBERT)				**68.8 / 80.6**
CA-AcdQA (BioBERT)				**73.2 / 83.8**
CA-AcdQA (CORD-19)				**80.6 / 87.9**

VI. ABLATION STUDY

We investigate the effect of the question expansion component and CNN Attention layer individually to understand the overall role they play in our model. In Table 4, we present our results without the CNN attention module in our model. Fine-tuning the QA pipeline without the CNN-Attention layer with pre-trained BERT and ALBERT reduced the performance significantly. Utilizing the CNN-Attention layer captures the semantic connections between the sentence and question features which boosts the model performance 7%-11% for EM and and F1 score. We depict the quantitative results in Table 5,

[7]https://huggingface.co/deepset

TABLE 4. The effect of CNN and Attention mechanism on the proposed models on all datasets.

Model	Tesla EM / F	California EM / F	EU-law EM / F	COVID-QA EM / F
CA-AcdQA(BERT) without CNN-Attention	82.3 / 85.0	81.1 / 84.7	80.0 / 84.2	45.0 / 66.4
CA-cdQA(BERT)	**89.8 / 93.2**	**89.5 / 92.8**	**88.0 / 92.2**	**55.6 / 76.4**
CA-AcdQA(ALBERT) without CNN-Attention	84.0 / 87.5	83.2 / 86.9	83.0 / 86.7	46.8 / 67.9
CA-AcdQA(ALBERT)	**92.6 / 95.5**	**92.3 / 95.3**	**91.2 / 95.4**	**57.5 / 78.8**

TABLE 5. The effect of question expansion component (QE) on the proposed models on all datasets.

Model	Tesla EM / F	California EM / F	EU-law EM / F	COVID-QA EM / F
CA-AcdQA(BERT) without QE	88.4 / 91.3	88.2 / 90.4	88.1 / 91.0	54.0 / 75.5
CA-AcdQA(BERT)	89.8 / 93.2	89.5 / 92.8	88.0 / 92.2	55.6 / 76.4
CA-AcdQA(ALBERT) without QE	90.9 / 93.5	91.1 / 93.0	90.8 / 94.6	57.2 / 77.6
CA-AcdQA(ALBERT) + T5	90.5 / 93.3	90.8 / 92.8	90.3 / 93.8	57.0 / 77.2
CA-AcdQA(ALBERT) + BART	91.2 / 94.0	91.1 / 93.6	91.0 / 94.8	57.2 / 77.7
CA-AcdQA(ALBERT)	**92.6 / 95.5**	**92.3 / 95.3**	**91.2 / 95.4**	**57.5 / 78.8**

where the model's performance with and without the question expansion (QE) component is shown. Additionally, we examined the BART and T5 pre-trained question paraphrasers for rewriting the vague questions. T5 caused a slight performance degradation compared to the "CA-AcdQA(ALBERT) without QE" since it is not tuned with any domain-specific training data for question paraphrasing. BART paraphraser outperformed T5, although it couldn't elevate the model performance significantly. T5 and BART need fine-tuning on a domain-specific paraphrase dataset for a better performance which is not feasible for every domain.

We can conclude that rewriting the question without considering the domain terminology misleads the model by generating domain irrelevant questions. Therefore, for generating in-domain question paraphrases (synonymized questions) without the need for training data for every domain, the proposed QE module operated well, and it improved the EM and F1 score by 1%-2% for all domains. We conducted additional experiments to study the role played by each question type, i.e., " What", "Where", "When", "Why", "Who", "How". Besides that, our goal is also to portray that the new customizations that we have made to Giveme5W1H are useful to our framework. This will allow us to get an overall

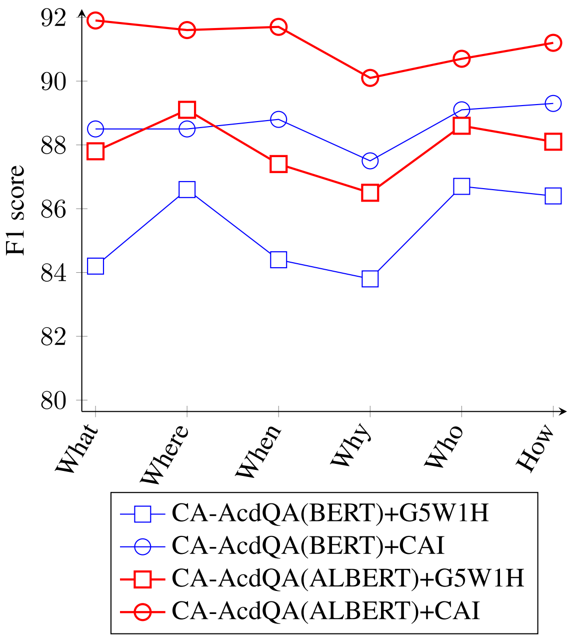


FIGURE 4. The average F1 score across all datasets for our model, CA-AcdQA (ALBERT), with the original Giveme5W1H and the proposed CAI on each question category.

understanding of the role that each question type plays in our study. We calculated the F1 score for each question type individually for analyzing our model performance on different question categories (see Table 6). We have also reported the number of instances in each question category on four datasets in Table 1. One observation is that the performance is not impacted by the number of instances in the category. We believe this is because the proposed framework does not heavily rely on statistical information, which makes it reliable even under low-resource situations. The number of questions that do not belong to the 5W1H categories is 15, 76, 59, 200, for Tesla, California, EU-law, and COVID-QA datasets. Furthermore, the candidate answer identifier helps automatically select the appropriate sentences in each question category based on the linguistic rules in III-A. An advantage that our model gets by the CAI component is reducing the number of candidate answers, which significantly impacts the model's effectiveness for long contexts by excluding the question-irrelevant sentences. Fig. 4 displays the average

TABLE 6. Performance comparison (F1) of our models for each question category with the proposed candidate answer identifier (customized Giveme5W1H) and Giveme5W1H candidate answer identifier.

		Proposed CAI (customized Giveme5W1H)						Giveme5W1H Candidate Answer Identifier					
Model	Dataset	What	Where	When	Why	Who	How	What	Where	When	Why	Who	How
CA-AcdQA(BERT)	Tesla	93.5	93.3	93.0	92.5	93.6	93.5	87.8	91.6	89.1	88.5	91.5	89.6
	California	92.7	92.1	93.0	91.5	93.5	94.2	87.9	89.2	88.7	87.4	91.4	90.0
	EU-law	92.8	92.1	92.5	91.2	92.8	92.0	89.0	91.3	88.2	88.4	90.0	90.7
	COVID-QA	75.3	76.7	76.9	74.9	76.6	77.7	72.3	74.5	71.8	70.9	74.2	75.5
CA-AcdQA(ALBERT)	Tesla	96.5	96.4	96.6	94.5	94.3	94.5	92.0	94.6	92.8	90.6	92.2	90.4
	California	96.6	95.9	95.0	94.3	94.2	95.6	92.1	92.8	90.6	90.2	92.8	91.3
	EU-law	95.9	95.7	95.6	94.4	95.8	95.0	92.4	93.3	91.7	92.2	93.1	93.8
	COVID-QA	78.9	78.6	79.7	77.2	78.5	79.9	75.0	75.7	74.6	73.1	76.3	77.0

F1 score across all datasets for the model with the original Giveme5W1H and the proposed CAI on each question category. We improved the model performance for each question category by adding linguistic rules and functions to Giveme5W1H, and the ''What'', ''Why'', and ''When'' categories improved the most.

VII. CONCLUSION

Our proposed closed-domain QA model improves upon state-of-the-art models across different closed-domain datasets: Tesla (person); California (region); EU-law (system); and COVID-QA (biomedical) dataset. We presented a novel approach by exploiting CNN and the self-attention mechanism to solve the generalization problem by training on small datasets. Our model calculates the semantic association between the extracted local features from context sentences and the question by employing CNN and the self-attention mechanism. Furthermore, components such as the candidate answers identifier and question expansion assist the model by limiting the choice to relevant sentences for each question category and removing ambiguity in questions by replacing some keywords. Experimental results illustrate that our proposed model outperforms different models on different domains without any knowledge base. In the future, our goal is to extend this model to be unified with a retriever to further improve our model for QA.

REFERENCES

[1] A. Andrenucci and E. Sneiders, ''Automated question answering: Review of the main approaches,'' in *Proc. 3rd Int. Conf. Inf. Technol. Appl. (ICITA)*, vol. 1, 2005, pp. 514–519.

[2] S. Liu, X. Zhang, S. Zhang, H. Wang, and W. Zhang, ''Neural machine reading comprehension: Methods and trends,'' *Appl. Sci.*, vol. 9, no. 18, p. 3698, Sep. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/18/3698

[3] G. Salton, ''The SMART system,'' in *Retrieval Results and Future Plans*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971.

[4] B. P. S. Rawat, W.-H. Weng, S. Y. Min, P. Raghavan, and P. Szolovits, ''Entity-enriched neural models for clinical question answering,'' 2020, *arXiv:2005.06587*.

[5] S. Badugu and R. Manivannan, ''A study on different closed domain question answering approaches,'' *Int. J. Speech Technol.*, vol. 23, no. 2, pp. 315–325, 2020.

[6] S. P. Lende and M. M. Raghuwanshi, ''Question answering system on education acts using NLP techniques,'' in *Proc. World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave)*, Feb. 2016, pp. 1–6.

[7] S. Sarkar, V. Rao, S. B. Mithra, and S. V. Rao, ''NLP algorithm based question and answering system,'' in *Proc. 7th Int. Conf. Comput. Intell., Modeling Simulation*, 2015, pp. 97–101.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' 2018, *arXiv:1810.04805*.

[9] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, ''Bidirectional attention flow for machine comprehension,'' 2016, *arXiv:1611.01603*.

[10] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang, ''SG-Net: Syntax-guided machine reading comprehension,'' in *Proc. AAAI*, 2020, pp. 9636–9643.

[11] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, ''SQuAD: 100,000+ questions for machine comprehension of text,'' 2016, *arXiv:1606.05250*.

[12] J. A. Alzubi, R. Jain, A. Singh, P. Parwekar, and M. Gupta, ''COBERT: COVID-19 question answering system using BERT,'' *Arabian J. Sci. Eng.*, pp. 1–11, Jun. 2021, doi: 10.1007/s13369-021-05810-5.

[13] Y. Lin, H. Ji, Z. Liu, and M. Sun, ''Denoising distantly supervised open-domain question answering,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1736–1745.

[14] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, ''End-to-end open-domain question answering with,'' in *Proc. Conf. North Amer. Assoc. Comput. Linguistics*, 2019, pp. 72–77.

[15] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, ''Dense passage retrieval for open-domain question answering,'' 2020, *arXiv:2004.04906*.

[16] M. Seo, J. Lee, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi, ''Real-time open-domain question answering with dense-sparse phrase index,'' 2019, *arXiv:1906.05807*.

[17] C. Qu, L. Yang, C. Chen, M. Qiu, W. B. Croft, and M. Iyyer, ''Open-retrieval conversational question answering,'' in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 539–548.

[18] D. Godavarthi and A. M. Sowjanya, ''Queries related to COVID-19: A more effective retrieval through finetuned Albert with BM25L question answering system,'' *World J. Eng.*, vol. 19, no. 1, pp. 109–113, Feb. 2022.

[19] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, ''ALBERT: A lite BERT for self-supervised learning of language representations,'' 2019, *arXiv:1909.11942*.

[20] L.-Q. Cai, M. Wei, S.-T. Zhou, and X. Yan, ''Intelligent question answering in restricted domains using deep learning and question pair matching,'' *IEEE Access*, vol. 8, pp. 32922–32934, 2020.

[21] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, ''Learning to retrieve reasoning paths over Wikipedia graph for question answering,'' 2019, *arXiv:1911.10470*.

[22] S. Min, D. Chen, L. Zettlemoyer, and H. Hajishirzi, ''Knowledge guided text retrieval and reading for open domain question answering,'' 2019, *arXiv:1911.03868*.

[23] T. N. Kipf and M. Welling, ''Semi-supervised classification with graph convolutional networks,'' 2016, *arXiv:1609.02907*.

[24] D. Chen, A. Fisch, J. Weston, and A. Bordes, ''Reading Wikipedia to answer open-domain questions,'' in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1870–1879.

[25] P. Qi, X. Lin, L. Mehr, Z. Wang, and C. D. Manning, ''Answering complex open-domain questions through iterative query generation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 2590–2602.

[26] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, ''Generation-augmented retrieval for open-domain question answering,'' in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4089–4100.

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

[28] R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum, ''Multi-step retriever-reader interaction for scalable open-domain question answering,'' in *Proc. Int. Conf. Learn. Represent.*, 2018.

[29] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder–decoder for statistical machine translation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[30] W. Xiong, X. Lorraine Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, W.-T. Yih, S. Riedel, D. Kiela, and B. Oğuz, ''Answering complex open-domain questions with multi-hop dense retrieval,'' 2020, *arXiv:2009.12756*.

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ''RoBERTa: A robustly optimized BERT pretraining approach,'' 2019, *arXiv:1907.11692*.

[32] F. Hamborg, C. Breitinger, and B. Gipp, ''Giveme5W1H: A universal system for extracting main events from news articles,'' 2019, *arXiv:1909.02766*.

[33] A. X. Chang and C. D. Manning, ''SUTIME: A library for recognizing and normalizing time expressions,'' in *Proc. LREC*, 2012, pp. 3735–3740.

[34] J. Pennington, R. Socher, and C. Manning, ''GloVe: Global vectors for word representation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[37] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
[38] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.
[39] L. Derczynski, J. Wang, R. Gaizauskas, and M. A. Greenwood, "A data driven approach to query expansion in question answering," 2012, *arXiv:1203.5084*.
[40] T. Möller, A. Reina, R. Jayakumar, and M. Pietsch, "COVID-QA: A question answering dataset for COVID-19," in *Proc. 1st Workshop NLP COVID-19 ACL*, 2020, pp. 110–111.
[41] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.
[42] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, and K. Toutanova, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, no. 29, pp. 453–466, 2019.
[43] D. Liu, Y. Gong, J. Fu, Y. Yan, J. Chen, D. Jiang, J. Lv, and N. Duan, "RikiNet: Reading Wikipedia pages for natural question answering," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6762–6771.
[44] J. Bergstra, D. Yamins, and David D. Cox, "Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms," *Comput. Sci. Discovery*, vol. 8, no. 1, 2013, Art. no. 014008.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[46] L. Xiao, N. Wang, and G. Yang, "A reading comprehension style question answering model based on attention mechanism," in *Proc. IEEE 29th Int. Conf. Appl.-Specific Syst., Architectures Processors (ASAP)*, Jul. 2018, pp. 1–4.
[47] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "QANet: Combining local convolution with global self-attention for reading comprehension," 2018, *arXiv:1804.09541*.
[48] Z. Zhang, J. Yang, and H. Zhao, "Retrospective reader for machine reading comprehension," 2020, *arXiv:2001.09694*.
[49] A. Otegi, J. A. Campos, G. Azkune, A. Soroa, and E. Agirre, "Automatic evaluation vs. User preference in neural textual QuestionAnswering over COVID-19 scientific literature," in *Proc. 1st Workshop NLP COVID EMNLP*, 2020, pp. 149–154.
[50] R. G. Reddy, B. Iyer, M. A. Sultan, R. Zhang, A. Sil, V. Castelli, R. Florian, and S. Roukos, "End-to-end QA on COVID-19: Domain adaptation with synthetic training," 2020, *arXiv:2012.01414*.
[51] S. Levy, K. Mo, W. Xiong, and W. Y. Wang, "Open-domain question-answering for COVID-19 and other emergent domains," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2021, pp. 259–266.
[52] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3615–3620.
[53] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020.
[54] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. M. Kinney, Z. Liu, W. Merrill, and P. Mooney, "CORD-19: The COVID-19 open research dataset," in *Proc. 1st Workshop NLP COVID-19 ACL*, 2020, pp. 1–12.

MAHSA ABAZARI KIA received the B.S. degree in computer science and the M.S. degree in software engineering from the University of Isfahan, Isfahan, Iran, in 2014 and 2018, respectively. 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 (Member, IEEE) received the B.Eng. and M.Eng. degrees in telecommunications from Kazan State Technical University, Russia, in 2007 and 2009, respectively, and the M.Sc. degree in communications and signal processing from Newcastle University, U.K., in 2009. She is currently a Research Manager of applied research with BT, U.K. Her current research interests include text analytics, natural language processing, and machine learning applied to desk-based operations improvement.

MATHIAS KERN received the M.Sc. and Ph.D. degrees in computer science from the University of Essex, U.K., in 1998 and 2006, respectively. He is currently a Senior Research Manager of sustainable resource management and optimization with the Applied Research Team, BT, U.K. 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 the OR and Analytics Forum (HORAF).

JON CHAMBERLAIN is currently a Senior Lecturer working with the School of Computer Science and Electronic Engineering, 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 Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. He is currently Lecturer with the Department of Electronics and Computer Science, University of Southampton. 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. He is an Associate Editor of *AI Communications* journal and has been recently awarded the Outstanding Early-Career Research Runner-Up Award by the Faculty of Science and Health at the University of Essex.