Disentangled Retrieval and Reasoning for Implicit Question Answering

Qian Liu, Xiubo Geng, Yu Wang, Erik Cambria, *Fellow, IEEE*, and Daxin Jiang

***Abstract*—To date, most of existing open-domain question answering methods focus on *explicit* questions where the reasoning steps are mentioned *explicitly* in the question. In this paper, we study *implicit* question answering where the reasoning steps are not evident in the question. Implicit question answering is challenging in two aspects. First, evidence retrieval is difficult since there is little overlap between a question and its required evidence. Second, answer inference is difficult since the reasoning strategy is latent in the question. To tackle implicit question answering, we propose a systematic solution denoted as DisentangledQA, which disentangles topic, attribute, and reasoning strategy from the implicit question to guide the retrieval and reasoning. Specifically, we disentangle topic and attribute information from the implicit question to guide evidence retrieval. For answer reasoning, we propose a disentangled reasoning model for answer prediction based on retrieved evidence as well as the latent representation of the reasoning strategy. The disentangled framework empowers each module to focus on a specific latent element in the question, and thus leads to effective representation learning for them. Experiments on the StrategyQA dataset demonstrate the effectiveness of our method in answering implicit questions, improving performance in evidence retrieval and answering inference by 31.7% and 4.5% respectively, and achieving the best performance on the official leaderboard. In addition, our method achieved best performance on the challenging EntityQuestions dataset, indicating the effectiveness in improving general open-domain question answering task.**

***Index Terms*—Natural Language Processing, Question Answering, Machine Reading Comprehensive.**

I. Introduction

OPEN-domain multi-step question answering (QA) [1, 2] is the task of answering questions by reasoning over multiple pieces of evidence which are retrieved from a large-scale corpus (e.g., Wikipedia). Typical open-domain QA methods are based on the *retriever-reader* paradigm [1, 3], in where the *retriever* to select evidence with the goal to cover the full required evidence, and a *reader* built on pre-trained language models to infer the final answer by jointly considering multiple pieces of evidence [4, 5, 6].

However, a key limitation of existing methods is that they only addressed *explicit* question answering where the reasoning process is mentioned explicitly in the question. For example, to answer question "*Is the area of Persian Gulf smaller than New Jersey?*" as shown in Fig. 1, the reasoning process is to retrieve the *area* of *Persian Gulf* and *New Jersey*, then infer the answer by applying the reasoning strategy of *size comparison*.

Qian Liu and Erik Cambria are with School of Computer Science and Engineering (SCSE), Nanyang Technological University, Singapore. Xiubo Geng and Daxin Jiang are with Microsoft STCA (Software Technology Center at Asia). Yu Wang is with University of Chicago.

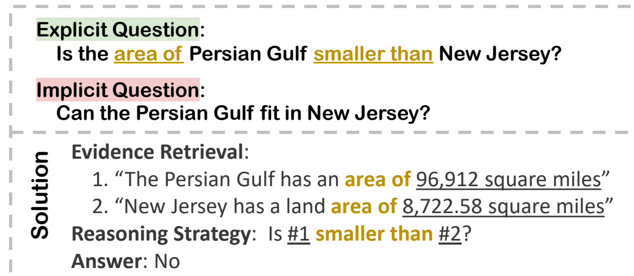


Fig. 1. Illustration of *explicit* question (*Q1*) and *implicit* question (*Q2*). They share the same pieces of evidence and reasoning strategy, which are explicitly mentioned in *Q1* (i.e., *area of* and *smaller than*) while this is implicit in *Q2*.

This reasoning process is expressed clearly (i.e., *the area of* and *smaller than*) in the question, which effectively guides the retrieval and reasoning. In reality, the reasoning process is often *implicit* in the question. For example, the implicit question "*Can the Persian Gulf fit in New Jersey?*" requires same reasoning strategy but without clues to retrieve *area* information and infer the answer by *comparison*. Due to implicit reasoning strategy, existing methods have failed in answering implicit questions and lag far behind their explicit counterparts on both retrieval and reading, with about 50% and 7% performance drop (as shown in Fig. 2), respectively.

The performance of existing methods on implicit QA is hindered by two major challenges. The first challenge is the evidence retrieval from the scale corpus with implicit and incomplete query information. For example, as shown in Fig. 3, to answer "*Can the Persian Gulf fit in New Jersey?*", both lexical and neural retrievers selected sentences about the *Persian Gulf* and *New Jersey* but failed to find the correct evidence about *area*. The main reason for this is that the *topics* (i.e., *New Jersey* and *Persian Gulf*) are explicitly mentioned but the required *attribute* (i.e., *area of*) is not. Another challenge is inefficient answer reasoning due to implicit strategies. Even when the golden evidence is provided, it is still challenging for the QA model to infer the correct answer without knowing the reasoning strategy (i.e., *size comparison*).

In this work, we present a new solution for answering implicit questions, denoted as DisentangledQA, which disentangles topic, attribute, and strategy from an implicit question to guide the evidence retrieval and reasoning. For the first challenge of evidence retrieval, our disentangled retriever consists of 1) a retriever to recall topic-related evidence, and 2) a retriever, which masks the topics in question and encodes the masked question as a latent query to further retrieve relevant attributes.

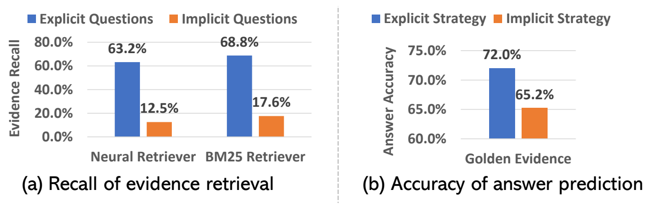


Fig. 2. Comparison of existing open-domain QA methods in answering explicit and implicit questions in terms of evidence retrieval and answer prediction. Neural retriever denotes DPR method [5]. The explicit questions and implicit questions are from Open-SQuAD [3, 7] dataset and StrategyQA [8] dataset, respectively.

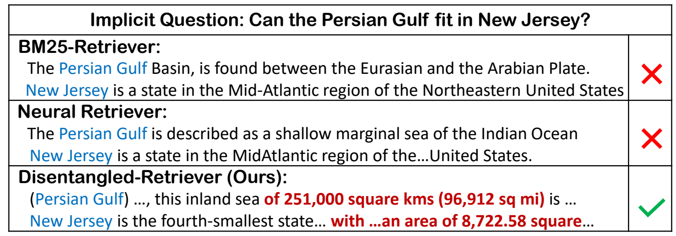


Fig. 3. Comparison of different retrieval methods for implicit question. Topic-related words are marked in blue and attribute-related words are marked in red.

The motivations of designing disentangle retriever are as follows: a) each candidate evidence piece in the open-domain corpus is about specific *attributes* of a *topic*; b) the required topics are usually mentioned explicitly while attributes are latent in implicit questions; and c) masking explicit topics makes it easier to infer the underlying attributes, for example answering *"Can X fit it Y"* requires *area* information.

For the second challenge, unlike the previous methods that only predict the answer using the retrieved evidence, our disentangled reasoning model first predicts the reasoning strategy with the masked question and masked evidence, and the final answer is predicted through the perception of the potential reasoning strategy. The key intuition motivating our design is that humans can easily judge that the question like "*Can X fit in Y?*" can be answered by *size comparison* over the evidence of *area of X* and *area of Y*.

The proposed disentangled retrieval and reasoning approach offers two benefits for open-domain QA. First, the disentangled information enables the model to focus on implicit attributes/reasoning strategy without being disturbed by explicit topics. Second, the disentangled retrieval and reasoning models employ separate modules for the explicit and implicit components of a question, which alleviates the learning difficulty of entangled questions.

In experiments, we first verify the effectiveness of our method on implicit questions. Then, we demonstrate our method is effective when applied to general open-domain QA task. More detailed, experiments on the StrategyQA [8] dataset (which is currently the only QA dataset for implicit questions) show that our method significantly outperforms previous methods for both evidence retrieval and QA by 31.7% and 4.5% respectively, achieving the best performance on the official leaderboard. Experiments on a challenging dataset, i.e., EntityQuestions [9], show that our method achieved the best performance than existing spare retrievers and dense retrievers, demonstrating the generalizability of our method on open-domain QA tasks.

We summarize our main contributions as follows:

- We highlight the importance of disentangling topic, attribute, and reasoning strategy from the implicit questions. The disentangled information helps to mine latent reasoning strategy from the question and guide the evidence retrieval and answer inference. To the best of our knowledge, this is the first work to tackle the problem of implicit QA.
- We design a disentangled evidence retrieval method which contains a topic retriever and an attribute retriever, which is effective for open-domain QA tasks.
- We design a disentangled reasoning method for answer inference by modeling the reasoning strategy under the implicit question.
- We conduct extensive experiments to evaluate the proposed method on the implicit QA dataset and the entity-centric QA dataset, showing superior performance over the state-of-the-art methods.

Code and data are available on our Github[1]. The rest of this paper is structured as follows: Section II discusses related researches about open-domain question answering; Section III introduces the problem formulation of implicit question answering and describes the details of the proposed DisentangledQA method, including disentangled retrieval and disentangle reasoning; Section IV compares our method and other baselines and provides in-depth analysis of the proposed method; finally, Section V offers concluding remarks.

II. Related Works

Open-domain QA is a task of answering questions from a large collection of documents, and its typical solution is the *retriever-reader* approach [1, 10, 11, 12, 13, 14], where a *retriever* searches a small set of question-related evidence from an open-domain corpus, then a *reader* forms the answer from the candidate evidence. In this section, we introduce the related works on the retrieval and reading components.

A. Evidence Retrieval

In the retriever-reader paradigm, the recall of the retriever significantly affects the final QA performance. Traditional methods [3] leverage *sparse* methods like TF-IDF [15] and BM25 [16, 17] to retrieve candidates from the evidence collection. However, they mainly rely on lexical matching and suffer from the term mismatching problem.

To further improve retrieval performance, *dense* retrieval methods [18, 19, 20, 21] are widely explored to encode text as dense vectors and retrieve evidence pieces of which vectors are closest to the question vector. For example, Karpukhin et al. [5] proposed the Dense Passage Retriever with a dual encoder to learn dense representations of questions and passages.

[1]https://github.com/senticnet/DisentangledQA.

Das et al. [22] proposed a multi-step retriever to iteratively retrieve evidence pieces from multiple documents. Nie et al. [4] designed a dense semantic retriever using paragraph-level and sentence-level BERT models to select paragraphs from paragraphs retrieved by TF-IDF. Asai et al. [23] proposed Path Retriever which employs BERT as an encoder and recursively selects the best passage sequence on top of a hyperlinked passage graph. Mao et al. [24] proposed a generation-augmented retrieval for answering open-domain questions, which augments a query through text generation of heuristically discovered relevant contexts without external resources as supervision. Seo et al. [25] introduced query-agnostic indexable representations of document phrases that can drastically speed up open-domain QA.

Following Seo et al. [25], Lee et al. [26] proposed an effective method to learn phrase representations from the supervision of reading comprehension tasks, coupled with novel negative sampling methods. More recently, researchers also found that exiting retrieval methods fail to retrieve evidence for complex and challenging questions from open-domain corpus. For example, Sciavolino et al. [9] focused on the entity-centric questions and suggested to incorporate explicit entity memory into dense retrievers to help differentiate rare entities. For multi-hop questions, Yadav et al. [27] designed an unsupervised alignment-based iterative evidence retrieval method. However, these methods are mainly designed for explicit questions and are not sufficient for to implicit questions which have little overlap with their evidence.

B. Question Answering

QA is a challenging task because it requires a simultaneous understanding of the question and evidence [28, 29, 30, 31, 32, 33]. Previous works have developed a number of deep neural architectures. For example, in visual QA task, Yu et al. [34] developed a multi-modal factorized bilinear pooling approach to understand the visual content of images and the textual content of questions. Yu et al. [35], designed co-attention learning to model both the image attention and the question attention simultaneously, to reduce the irrelevant features effectively.

Recently, pre-trained language models such as BERT [36] and RoBERTa [37] have become the typical *readers* for QA systems. Benefiting from pretraining and powerful transformers for capturing the contextualized representations [37, 38, 39], these methods achieved state-of-the-art QA performance, especially for questions where the answer is explicit in a single evidence piece [7, 40]. To answer questions with multi-step reasoning, researchers proposed decomposing the question into several sub-questions and conduct retrieval and reasoning for multiple steps. For example, Min et al. [41] proposed a system for multi-hop method that decomposes a compositional question into simpler sub-questions that can be answered by off-the-shelf single-hop models. Perez et al. [42] designed an One-to-N unsupervised sequence transduction that learns to map one hard, multi-hop question to many simpler, single-hop sub-questions.

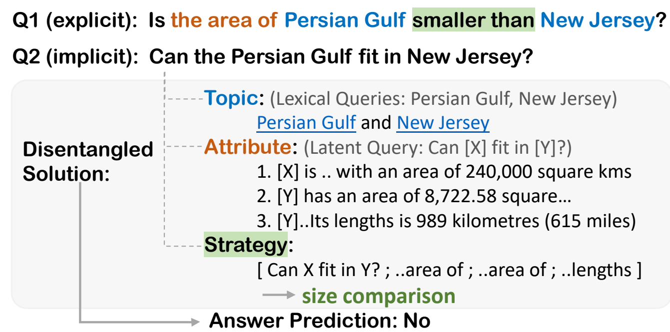


Fig. 4. Illustration of the proposed method for answering an implicit question. Q1 and Q2 are the same questions in their explicit and implicit expressions, respectively. To answer the implicit question Q2, our disentangled solution is to disentangle the topic, attribute, attribute, and strategy from the question, then jointly infer the answer.

Wolfson et al. [43] introduced a question decomposition meaning representation (QDMR) for questions, which constitutes the ordered list of steps, expressed through natural language, that are necessary for answering a question. Lewis et al. [44] proposed a pretrained sequence-to-sequence method BART, which is able to decompose the question into several sub-questions. Cheng et al. [12] designed a hybrid approach for leveraging both extractive and generative readers, and found that proper training methods can provide large improvement over previous models. Pan et al. [45] proposed an unsupervised framework that can generate human-like multi-hop training data from both homogeneous and heterogeneous data sources.

However, these methods fail to answer implicit questions. The required reasoning steps are unclear, and this makes it difficult to reasonably decompose the question or explore QA shortcuts [40] using transformers. In this work, we proposed a disentangled solution to answer implicit questions. It has been widely studied in cross-modality visual QA for the idea of disentangling reasoning. For example, Yi et al. [46] presented a neural-symbolic approach for visual QA that disentangles reasoning from visual perception and language understanding. Yi et al. [47] introduced a dataset named CLEVRER for systematic evaluation of computational models on a wide range of reasoning tasks. Chen et al. [48] designed a unified neural symbolic framework named Dynamic Concept Learner to study temporal and causal reasoning in videos. Following this line, we designed a disentangled solution to answer implicit questions.

To sum up, unlike previous methods, our method is designed to answer implicit questions. We disentangle the topic and attribute information from the question to retrieve concise evidence and disentangle a latent reasoning strategy for answer inference.

III. Methodology

In this section, we first introduce the overview of the proposed method and then detail each module, i.e., disentangled retrieval and disentangled reasoning.

A. Overview

Implicit QA takes a natural language question q as input, with the goal of forming the answer using an open-domain corpus $\mathcal{C}$, which contains large-scale documents on diverse topics. The reasoning strategy to infer the answer is implicit in the question q. Generally, a *retriever* is first designed to collect a small set of candidate evidence pieces $\mathcal{E}_q$ over the large-scale open-domain corpus $\mathcal{C}$. Then, a *reader* is designed to form the answer with the question and pieces of evidence in $\mathcal{E}_q$. There are two metrics to evaluate the task performance, i.e., 1) $\mathbf{Recall@10}$ is the fraction of golden paragraphs in the top-10 paragraphs generated by the retriever; and 2) $\mathbf{Accuracy}$ is the percentage of questions where the answer is correctly predicted by the reader.

The difficulty in answering implicit questions is that there is no mention of the reasoning steps and strategy, which poses the combined challenge of retrieving the relevant context and deriving the answer based on that context. To solve this problem, we propose to disentangle topic, attribute and reasoning strategy from the question to guide retrieval and reasoning. We illustrate the proposed DisentangledQA method to answer the implicit question *"Can the Persian Gulf fit in New Jersey?"* in Fig. 4. Specifically, our method highlights:

- **topic** information is explicitly mentioned in question, e.g., *Persian Gulf* and *New Jersey*, which is an important clue to retrieve relevant documents from the open-domain corpus;
- **attribute** information is the required aspects of *topics* to answer the question, e.g., *area of* of *Persian Gulf*, which is hidden in the question and we model it as latent query to search concise sentences from the topic-related documents;
- **reasoning strategy** is the operation to infer answer with the question and evidence, such as *size comparison* to answer *Can X fit in Y* with the evidence of *the area of X* and *the area of Y*.

With this disentangled solution, our disentangled retrieval consists of 1) a topic retriever to search topic-related evidence; and 2) an attribute retriever to search concise sentences of evidence. Our disentangled reasoning module consists of 1) a strategy predictor to infer the latent reasoning strategy; and 2) an answer predictor to infer the answer with question, evidence, and latent reasoning strategy.

B. Disentangled Retrieval

The disentangled retrieval method (denoted as Disentangle Retriever) contains a topic retriever and an attribute retriever to select evidence for question answering.

1) Topic Retriever: The topic retriever first generates a small set of documents $\mathcal{D}_q = \{d_1, d_2, \cdots, d_n\}$ which are topical-related to question q. To completely cover the topic information of the question, we design a multi-view query generator to obtain queries from the question:

- Named-entity recognition (NER): a pre-trained NER model[2] is used to extract the named entities (such as person names and locations) from the question.
- Nouns: the noun words and phrases in the question, which are identified by the part-of-speech tags[3].
- N-Grams: the unigram, bigram, trigram, and so on to the n-grams of the question, where n is the length of the question[4]. Considering the huge number of n-grams, we use exact matching in the retrieval process to avoid noise.

All these queries are combined as a query set and used to search documents from the open-domain corpus $\mathcal{C}$ using the BM25 function [17]. We search the topic-related fields of $\mathcal{C}$ (e.g., titles of Wikipedia pages or news). All documents in $\mathcal{C}$ are indexed by their titles. We combine the top-50 documents of all queries and rerank them with a RoBERTa-based classifier [37], where the input sequence is the concatenated question and document title. Top-n ($n \ll |\mathcal{C}|$) documents with maximum probability are selected as candidate document set $\mathcal{D}_q$.

As suggested by Min et al. [49], most questions can be answer by a small set of sentences. The topic-related documents in $\mathcal{D}_q$ contain a large amount noise sentences. To avoid interference by noise information, we train a paragraph-level classifier to filter out irrelevant context. Specifically, all documents in $\mathcal{D}_q$ are split into paragraphs. The question-aware paragraph representation is obtained as follows:

$$\mathbf{h}_{para} = \text{Transformer}(\texttt{[CLS]}\, q\, \texttt{[SEP]}\, para), \tag{1}$$

where *Transformer* denotes a pre-trained language model where the input sequence is the concatenation of question q and candidate paragraph $para$, and $\mathbf{h}_{para}$ is the representation of `[CLS]` which is pre-trained to summarize the latent meaning of the input sequence. Then, it is fed into an output layer for classification:

$$p^{(t)} = \text{sigmoid}(\text{FFN}(\mathbf{h}_{para}; \theta)), \tag{2}$$

where $\text{FFN}(\cdot; \theta)$ denotes a θ-parameterized one-layer feed-forward network, and $p^{(t)}$ is the probability distribution. The training objective is designed as:

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{N} (y^{(t)} \log p^{(t)} + (1 - y^{(t)}) \log(1 - p^{(t)})), \tag{3}$$

where N is the number of question-paragraph pairs, $y^{(t)}$ is the label which is set to 1 when the paragraph contains the evidence, and 0 otherwise.

With the trained classifier, we can evaluate the score of each testing question-paragraph pair, since $p^{(t)}$ indicates the paragraph is relevant or irrelevant to the topic of question. The top ranked paragraphs to the question are selected by thresholding the number of selected paragraphs, where the threshold is a hyperparameter.

[2]We use a BERT-large-cased model fine-tuned on CoNLL-2003, which is available on https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english.

[3]We use the NLTK toolkit and the nouns are labeled by NN, NNS, NNP, or NNPS, i.e., https://www.nltk.org/book/ch05.html.

[4]We use the `everygrams` function in NLTK to generate n-grams, i.e., https://www.nltk.org/api/nltk.html.

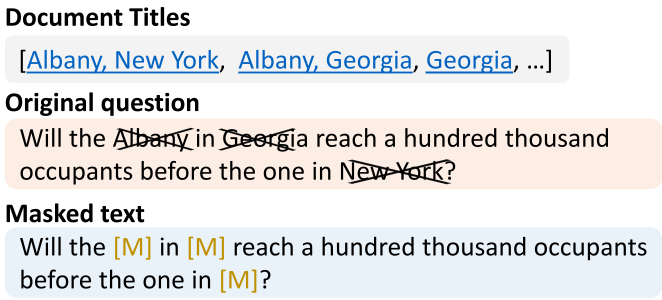


Fig. 5. Illustration of the mask mechanism. [M] denotes a blank character. *Documents Titles* are examples of the searched titles by Topic Retriever. For the *original question*, we replace the topic-related words (e.g., *Albany* and *New York*) using [M]. *Masked text* denotes the masked question.

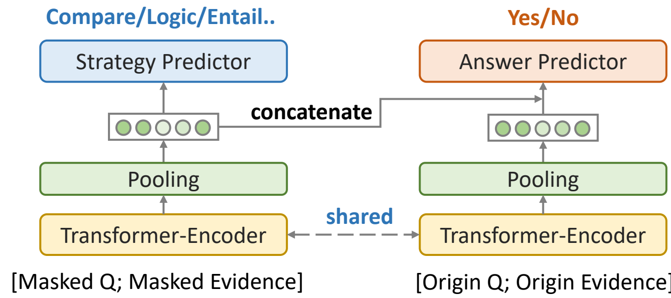


Fig. 6. Illustration of disentangled reasoning method, which contain a strategy predictor and an answer predictor. These two predictors have different input sequences with shared encoder. The strategy predictor is to predict implicit reasoning strategy. The answer predictor is to predict the answer to the question.

We split the selected paragraphs into sentences and generate a small set of candidate sentences $\mathcal{E}_q^t = \{s_1, s_2, \cdots, s_m\}$, which contains topic-related information to the question q.

2) Attribute Retriever: Given the candidate set $\mathcal{E}_q^t$ which are topic-related to question q, the attribute retriever is designed to retrieve a small set of sentences $\mathcal{E}_q^a$ which are true evidence with required attribute information to answer the question.

Intuitively, the attribute (e.g., *area of*) is the key guide to find true evidence from various sentences which describe the topics. However, it is implicit in q. To alleviate this problem, considering the question in Fig. 4, we assume that the attribute *area of* is hidden in *fit in*, and employ a mask mechanism and a deep encoder to map the question and evidence into a vector space, where the potential associations between *fit in* and *area of* can be captured by vector similarity.

First, a mask mechanism is designed to help the retriever focus on the part of q that implies attribute information, rather than being distracted by explicit topics. As shown in Fig. 5, we create a mask word set $\mathcal{M}_q$ which contain words in the document titles in $\mathcal{D}_q$. Stop words are removed from $\mathcal{M}_q$. We mask question q by removing these mask words:

$$q^* = \{q_i|_1^{|q|}, q_i \notin \mathcal{M}_q\}, \tag{4}$$

where q_i is a word in question q and q^* is the masked question. Similarly, the mask mechanism converts each sentence s_i in $\mathcal{E}_q^t$ as its masked version s_i^*.

Then, the attribute retriever applies a dense encoder $Enc(\cdot)$ to map any text into a fixed-size dense vector. We follow Sentence-Transformer [50] to add a pooling operation to the output of RoBERTa to embed input text as a vector. All the masked sentences are represented as dense vectors and indexed into a vector search space. Then, the masked question is encoded as a query vector to search the top-k sentences of which vectors are the closest to the query vector. We employ the `MEAN` pooling strategy and the similarity of each sentence s_i to question q, which is computed using dot product:

$$sim(q, s_i) = Enc(q^*)^\top \cdot Enc(s_i^*). \tag{5}$$

The training objective is to fine-tune the encoder so that relevant pairs of questions and sentences have a higher similarity than the irrelevant ones. For example, *Can X fit in Y* is closer to *the area of X/Y* than *the history of X/Y*.

The training sample contains a question q, a positive evidence sentence s^+, and n negative sentences $\{s_1^-, s_2^-, \cdots, s_l^-\}$ randomly selected from $\mathcal{E}_q^t$, and we optimize the loss function as:

$$\mathcal{L}_{enc} = \sum^N - \log \frac{e^{sim(q,s^+)}}{e^{sim(q,s^+)} + \sum_{j=1}^l e^{sim(q,s_j^-)}}, \tag{6}$$

where N is the size of the training samples.

3) Data Augmentation: It is expensive to search or label gold evidence sentences for implicit questions. According to the statistics of Geva et al. [8], the human performance in finding a gold paragraph without question decomposition is only 51.3% in terms of recall. To train a robust encoder, we use multiple rounds of training and use the retrieval results of the last round as the pseudo-label data to carry out data augmentation. First, we train the encoder using the labeled sentences as positive examples, and randomly select negative sentences from the documents. Then, of the top-k similar sentences to a question, sentences from the gold paragraphs are used as pseudo positive data and the others as pseudo negative data. The pseudo data is used to fine-tune the encoder in the next round.

C. Disentangled Reasoning

Given question q and retrieved evidence sentences $\mathcal{E}_q = \{s_1, s_2, \cdots, s_k\}$, the disentangled reasoning method attempts to form the answer by understanding the implicit strategy. As shown in Fig. 6, our disentangled reasoning model contains 1) a strategy predictor to learn the latent reasoning strategy of the question and 2) an answer predictor to conduct a strategy-aware answer inference.

1) Reasoning Strategy Predictor: Intuitively, the reasoning strategy is latent in the masked question and evidence sentences. For example, given the question *"Did X fit in Y"* with the several evidence sentences *"the area of X is..."* and *" Y has the area of ..."*, the predictor is expected to infer that the reasoning strategy is *size comparison*.

In our method, we train a reasoning strategy predictor based on the pre-trained language model. The masked question and evidence sentences are concatenated as input sequence:

$$\mathbf{h}^* = \text{Transformer}(\texttt{[CLS]}\, q^* \,\texttt{[SEP]}\, s_1^*, s_2^*, \cdots, s_k^*), \tag{7}$$

where the *Transformer* denotes the pre-trained language model with a pooling layer to convert the input sequence as a fixed-length vector $\mathbf{h}^*$. Here, we employ the representation of `[CLS]` as the pooling method, which is pre-trained to summarize the latent meaning of the input sequence. Then, a reasoning strategy predictor is built to predict the reasoning strategy using a neural classifier:

$$\mathbf{p}^{(s)} = \text{softmax}(\text{FFN}(\mathbf{h}^*;\theta)), \tag{8}$$

where $\text{FFN}(\cdot;\theta)$ denotes a θ-parameterized one-layer feed-forward network, and $\mathbf{p}^{(s)}$ is the probability distribution of the reasoning strategy types. In our method, the strategy of the training data is annotate by a keyword matching method (as detailed in Section IV-A). The predictor is trained by minimizing the negative log probability of the ground-truth strategy label:

$$\mathcal{L}_{strategy} = -\frac{1}{N}\sum_{N}\sum_{i=1}^{C}\mathbf{y}_i^{(s)}\log\mathbf{p}_i^{(s)}, \tag{9}$$

where $\mathbf{y}^{(s)}$ is the one-hot representation of the strategy type labels, C is the number of types, and N is the number of training samples.

2) Answer Predictor: We leverage the latent reasoning strategy to help the answer inference. First, we learn the latent question-evidence representation $\mathbf{h}$ based on the pre-trained language model:

$$\mathbf{h} = \text{Transformer}(\texttt{[CLS]}\,q\,\texttt{[SEP]}\,s_1, s_2, \cdots, s_k), \tag{10}$$

where *Transformer* is the shared encoder with reasoning strategy predictor. We concatenate it with latent vector $\mathbf{h}^*$ to infer the answer. For the boolean answer (i.e, *yes* or *no*), we employ the binary classifier with the sigmoid function to predicate the answer:

$$p^{(a)} = \text{sigmoid}(\text{FFN}(\mathbf{h}\oplus\mathbf{h}^*;\theta)), \tag{11}$$

where $\text{FFN}(\cdot;\theta)$ denotes a θ-parameterized one-layer feed-forward network, and $p^{(a)}$ is the probability distribution of answers. It is trained by minimizing the negative log probability of the ground-truth strategy label:

$$\mathcal{L}_{ans} = -\frac{1}{N}\sum_{N}(y^{(a)}\log p^{(a)} + (1-y^{(a)})\log(1-p^{(a)})), \tag{12}$$

where $y^{(a)}$ is the ground-truth answer label which is set to 1 when the answer is *yes*, and 0 otherwise. N is the number of training samples.

We jointly train the reasoning strategy predictor and the answer predictor:

$$\mathcal{L} = \mathcal{L}_{ans} + \lambda\mathcal{L}_{strategy}, \tag{13}$$

where λ is a combination parameter.

To sum up, **Algorithm 1** shows high-level pseudo-code for the DisentangledQA method in evidence retrieval and answer inference.

Algorithm 1: The DisentangledQA Method

Input: question q, open-domain corpus $\mathcal{C}$, epoch of data augmentation N
// Disentangled Retrieval
Step1: Topic Retriever
Generate queries with multi-view query generator;
Retrieve titles using BM25 retriever;
Generate $\mathcal{D}_q$ by re-ranking titles;
Select topic-related sentences $\mathcal{E}_q^t$ from $\mathcal{D}_q$ by Eq. (3);
Step2: Attribute Retrieval
Sample training samples S for each question in the training dataset;
for $i = 1$ **to** N **do**
  Optimize the attribute encoder with S by Eq. (6);
  Evaluate all candidate sentences using Eq. (5);
  Select pseudo data and add them into S;
end
Obtain $\mathcal{E}_q^a$ from $\mathcal{E}_q^t$ by Eq. (5);
// Disentangled Reasoning
Step3: Answer Inference
Train the strategy predictor and the answer predictor by Eq. (13);
Get $\mathbf{h}$ and $\mathbf{h}^*$ with $\mathcal{E}_q^a$ by Eq. (10) and Eq. (7);
Predict the answer by Eq. (11);
Output: bolean answer (*yes* or *no*)

TABLE I
STATISTICS OF THE STRATEGYQA. *# Question* IS THE NUMBER OF QUESTIONS, *Avg.Len* IS THE AVERAGE QUESTION LENGTH. *Avg.Doc* AND *Avg.Para* DENOTE THE AVERAGE NUMBER OF DOCUMENTS AND PARAGRAPHS TO ANSWER THE QUESTIONS, RESPECTIVELY. *%Yes* IS THE PERCENTAGE OF QUESTIONS WHOSE ANSWER IS *yes*.

StrategyQA	# Question	Avg.Len	Avg.Doc	Avg.Para	% Yes
Train	2,061	9.6	1.97	2.33	46.8%
Dev	229	9.7	1.95	2.30	46.7%
Test	490	9.8	-	2.29	46.1%

IV. EXPERIMENTS

In this section, we evaluate the effectiveness of our method. We first detail the experiment setting, including the dataset, implementation, and the compared baselines. Then, we compare the proposed method with different methods and show the overall performance, followed by the ablation study, in-depth analysis, and case study.

A. Datasets and Implementation

First, we evaluate the effectiveness of the proposed DisentangledQA in answering implicit questions using StrategyQA [8], which is a boolean QA dataset with implicit questions. To the best of our knowledge, this is the only implicit QA dataset with a variety of complex question answering strategies. It contains 2,290 question-answer pairs with annotated facts, evidence paragraphs and question decomposition for training and 490 questions for online testing. It also provides a 90%/10% split of training data to get the in-house training/development split.

TABLE II
KEYWORDS FOR DIFFERENT REASONING STRATEGIES.

Strategy	Keywords
comparison	greater, less, smaller, higher, lower, longer, shorter...
binary	same, identical, equal, different, difference, match...
numerical	least, times, plus, multiplied, divided, positive...
logical	or, all, also, both
entail	contain, absent, overlap, included, within, excluded...

The statistics of the dataset are shown in Table I. The corpus to answer the implicit questions in StrategyQA is an open-domain Wikipedia dump[5], which contains 5.98M Wikipedia documents with 36.6M processed paragraphs. The answer is *Yes* or *No*. In the training and development datasets, each implicit question is labeled with the evidence and reasoning strategy. Each example in the test dataset simply comprises a question, and the answer, evidence, and reasoning strategy are hidden. In the official evaluation, the participant methods are compared with the accuracy of answers and the recall of the top-10 retrieved paragraphs.

In our experiment, the topic retriever leverages the Python Elasticsearch API[6] to index all Wikipedia documents. In the topic retriever, the query for each question is multi-view queries designed in Section III-B1, the search domain is *Title*. We train the attribute retriever using a fine-tuned Sentence-Transformer[7] and set the parameters as follows: the sequence length is 128, the batch size is 256, the learning rate is 3e-5, and the number of training epochs is 10. The selected sentences are used for QA, and the paragraphs where these sentences are located are used to evaluate Recall@10. The disentangled reasoning model is built on RoBERTa*, which is a fine-tuned RoBERTa [37] model on DROP [51], 20Q[8], and BoolQ [52] by Geva et al. [8]. RoBERTa* is available online[9]. For the reasoning strategy annotation, we extract the last step of human-written question decomposition and perform keyword matching. There are five classes of reasoning strategies, i.e., *comparison*, *logical*, *entail*, *binary* and *numerical*. Table II shows several examples of the used keywords, and all of the used keywords are released[10]. We set the used parameters as follows: the batch size is 16, the sequence length is 512, the learning rate is 1e-5, the warm up rate is 0.1, and the number of training epochs is 5.

Second, to verify that our method generalizes well to open-domain QA task, we conduct experiments on the EntityQuestions [9] dataset, which contains 24 types of entity-centric questions. The open-domain corpus for answering these questions is also the Wikipedia dump. It is a challenging dataset for dense retrieval methods. As observed by Sciavolino et al. [9], the dense retrieval method (i.e., Dense Passage Retrieval [5]) drastically underperforms the sparse BM25 baseline (49.7% vs 72.0% on average), with the gap on some question patterns reaching 60% absolute. Note that EntityQuestions only contains explicit questions with 24 explicit strategies, such as *"Where was [E] born?"* and *"Where is [E] located?"* (*[E]* denotes an entity), and no implicit reasoning strategies are required. As such, we employ the proposed Disentangle Retriever on this dataset and compare our method with other state-of-the-art retrieval methods.

We setup the experiments on EntityQuestions following the official repository[11]. We also employ Python Elasticsearch API to index all Wikipedia documents for the topic retriever. Considering most of questions in EntityQuestions have formal entities, we employ a lexical classifier[12] to select top-5 documents, instead of a RoBERTa-based classifier. For training the attribute retriever, we fine-tune a Sentence-Transformer and set the parameters as follows: the sequence length is 128, the batch size is 256, the learning rate is 2e-5, and the number of training epochs is 3. In this experiment, we use the official evaluation metrics, i.e., top-20 retrieval accuracy.

B. Baselines

We compare our method with the following baselines. Traditional methods directly retrieve paragraphs from the whole Wikipedia corpus using BM25, then the question and the retrieved top-10 paragraphs are fed into a RoBERTa-based reader or a RoBERTa*-based reader to predict the answer. The used queries for retrieval include:

- **IR-Q** [8] uses a query that consists of the non-stop words of the original question.
- **IR-D** decomposes a question into several sub-questions using BART [44] and initiates a separate query for each decomposition. The retrieved paragraphs of all steps are sorted by their retrieval scores.

We design a topic retriever to select a small set of documents $\mathcal{D}_q$. We re-implement the following baselines based on our topic retriever.

- **IR-Q$^{\triangle}$** employs the BM25 function to select the top-10 paragraphs for each question from $\mathcal{D}_q$. All the selected paragraphs are concatenated as evidence.
- **Dense Passage Retrieval (DPR)** [5] employs a dual encoder to encode the questions and paragraphs as dense vectors and the top-10 paragraphs which are the closest to the questions are selected.
- **Joint Retrieval** jointly evaluates the evidence chain, following Yadav et al. [53]. In our implementation, any two paragraphs retrieved by DPR are joined as an evidence chain. A RoBERTa-based classifier is trained to select an evidence chain.
- **Semantic Retrieval** [4] is a multi-grained evidence retrieval method based on RoBERTa, which jointly considers the paragraph-level and sentence-level semantic matching to select the evidence.

[5]https://storage.googleapis.com/ai2i/strategyqa/data/corpus-enwiki-20200511-cirrussearch-parasv2.jsonl.gz
[6]https://github.com/eladsegal/strategyqa/tree/main/elasticsearch_index
[7]https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2
[8]https://github.com/allenai/twentyquestions
[9]https://storage.googleapis.com/ai2i/strategyqa/models/2_boolq.tar.gz
[10]https://github.com/senticnet/DisentangledQA/tree/main/classification.
[11]https://github.com/princeton-nlp/EntityQuestions
[12]We employ the `get_close_matches` function defined in difflib (https://docs.python.org/3/library/difflib.html).

TABLE III
OVERALL PERFORMANCE OF ALL THE METHODS ON THE DEVELOPMENT SET OF STRATEGYQA. $\triangle$ DENOTES EVIDENCE SELECTION FROM DOCUMENTS RETRIEVED BY OUR TOPIC RETRIEVER.

#	Methods	Recall@10	Accuracy
	Open-domain Corpus		
0	Human Performance	*58.6%*	*87.0*
1	MAJORITY	-	53.3
2	RoBERTa-IR-Q	18.2%	57.2
3	RoBERTa*-IR-Q	18.2%	62.4
4	RoBERTa*-IR-D	19.5%	65.5
5	RoBERTa*-IR-Q$^{\triangle}$	36.2%	63.3
6	RoBERTa*-DPR$^{\triangle}$	51.4%	64.2
7	RoBERTa*-Joint Retrieval$^{\triangle}$	51.6%	65.5
8	RoBERTa*-Semantic Retrieval$^{\triangle}$	48.4%	63.8
9	RoBERTa*-**Disentangled Retriever**$^{\triangle}$	**55.9%**	66.8
10	**DisentangledQA (Our)**	**55.9%**	**68.1**
	ORACLE Paragraphs		
11	RoBERTa*-ORA-P	-	70.7
12	RoBERTa*-ORA-P-D	-	72.0
13	**DisentangledQA (Our)**	-	**73.8**

TABLE IV
PERFORMANCE COMPARISON ON THE HIDDEN TESTING SET OF STRATEGYQA. † DENOTES THE PUBLISHED RESULT AND ‡ DENOTES THE RESULT REPORTED IN OFFICIAL LEADERBOARD.

Methods	Recall@10	Accuracy
MAJORITY†	-	53.9
ROBERTA*-∅ † [8]	-	63.6
DPR for retrieval‡	12.5%	-
RoBERTa-IR-Q† [8]	17.4%	53.6
RoBERTa*-IR-Q‡	17.3%	64.9
RoBERTa*-IR-D‡ [44]	17.4%	60.2
GPT-3	-	59.2
DisentangledQA	**48.9%**	**66.1**
DisentangledQA(ensemble)	**48.9%**	**69.4**

C. Overall Performance

1) Performance on StrategyQA: Table III summarizes the results of all the methods on the development dataset of StrategyQA. *MAJORITY* denotes the performance without training, and *ORACLE Paragraphs* denote the question answering with the golden paragraphs. The first group (#1-10) is the open-domain implicit QA. We observe that the proposed DisentangledQA achieves a significantly better performance than the other baselines, both on retrieval and QA, with an average performance gain of 21.1% and 4.9%, respectively. This observation indicates the effectiveness of our method in jointly leveraging topic, attribute and strategy information to answer implicit questions.

Focusing on evidence retrieval, IR-Q and IR-D achieve poor performance with an average recall of 18.6%, which affects the follow-up QA. When equipped with a topic retriever, IR-Q$^{\triangle}$ achieves 18% performance gain in terms of Recall@10, showing that the disentangled topic information from the question as a query is effective to reduce the search space. Moreover, a topic retriever also benefits the other dense retrievers (#6-8), with an average improvement of 28.3%. Our attribute retriever (#9) achieved the best retrieval performance, indicating the importance of attribute information in evidence selection.

In relation to QA accuracy, we observe that RoBERTa*-IR-Q substantially outperforms RoBERTa-IR-Q with a gain of 5.2%, indicating that fine-tuning on the related auxiliary datasets is crucial. Compared with our method (#10), it is observed that removing the strategy predictor (#9) leads to a 1.3% QA performance drop, indicating that understanding the implicit reasoning strategy is helpful to inference the answer. Moreover, considering the oracle setting with the golden paragraphs, we compare DisentangledQA with the RoBERTa* method, where the input sequence is ORA-P (concatenated golden paragraphs, #11) and ORA-P-D (concatenated golden evidence for decomposition sub-questions, #12), and we observe that our method (#13) achieves better performance, with a 3.1% and 1.5% improvement, respectively. This observation shows our method is effective, and it benefits from revealing the latent reasoning strategy in answering implicit questions.

A comparison of the different methods on the hidden testing dataset is shown in Table IV. The proposed DisentangledQA achieves state-of-the-art performance in the leaderboard, indicating its effectiveness.

2) Performance on EntityQuestions: Table V summarizes the overall performance of different methods on EntityQuestions dataset. We compare our Disentangle Retriever with sparse retriever (i.e., BM25) and dense retrievers (i.e., DPR and REALM [54]). More specific, DPR(NQ) denotes the DPR model trained on Nature Questions dataset [55], which is a large-scale extractive QA dataset, and DPR(multi) denotes the DPR model trained on four QA datasets (i.e., NQ, TriviaQA [56], WebQ [57], and TRECQA [58]) combined. REALM adopts a pre-training task called salient span masking (SSM), along with an inverse cloze task from Lee et al. [18]. We also evaluate the performance of BM25 and DPR based on our topic retriever, which are denoted as BM25$^{\triangle}$ and DPR$^{\triangle}$, respectively.

It is observed that our method achieves best performance, indicating our method generalizes well to explicit open-domain QA. The main advantage of our approach is to disentangle topics and attributes, which are denoted as *entity* and *question pattern* in EntityQuestions dataset, respectively. With a topic retriever, DPR$^{\triangle}$ outperforms DPR(NQ) and DPR(multi) by 26% and 19% on average, respectively, indicating that disentangling topics and attributes is helpful for dense retrievers. Our method achieves better performance than DPR$^{\triangle}$, with an average performance gain of 0.8%, indicating the effectiveness of attribute retriever. We observe that the improvement from attribute retriever in StrategyQA is more significant than that in EntityQuestions (i.e., 4.5% v.s. 0.8%). This shows that DPR can search evidence for explicit questions, but cannot deal with implicit questions. Our method can effectively retrieve the evidence of implicit questions.

D. Ablation Study

We conduct an ablation study on the development dataset to understand how components affect the results. The results are reported in Table VI. It is observed that removing topic retriever leads to 30% performance drop in terms of Recall@10, indicating the importance of generating a small set of topic-related sentences $\mathcal{E}^{(t)}$ from the whole corpus $\mathcal{C}$.

TABLE V
OVERALL PERFORMANCE OF DIFFERENT METHODS ON THE TEST SET OF ENTITYQUESTIONS IN TERMS OF TOP-20 RETRIEVAL ACCURACY. NUM. DENOTES THE NUMBER OF QUESTIONS IN DIFFERENT TYPE OF RELATIONS. BM25$^{\triangle}$ AND DPR$^{\triangle}$ DENOTE BM25 AND DPR BASED ON THE DOCUMENTS RETRIEVED BY OUR TOPIC RETRIEVER.

Questions	**Num.**	**BM25**	**DPR(NQ)**	**DPR(multi)**	**REALM**	**BM25$^{\triangle}$**	**DPR$^{\triangle}$**	**Our Method**
P106 What kind of work does [E] do?	1000	71.2	25.9	52.9	53.6	78.9	79.0	**79.3**
P112 Who founded [E]?	510	81.2	77.1	75.7	77.3	80.8	81.4	**82.5**
P127 Who owns [E]?	1000	78.4	60.7	63.8	73.6	78.2	79.6	**81.2**
P131 Where is [E] located?	1000	63.1	45.7	44.2	63.9	75.0	75.2	**75.3**
P136 What type of music does [E] play?	1000	48.7	37.4	36.8	42.6	52.7	53.2	**53.9**
P159 Where is the headquarter of [E]?	1000	85.0	70.0	72.0	70.4	84.9	85.7	**86.5**
P17 Which country is [E] located in?	1000	61.5	64.2	67.7	70.6	69.0	69.3	**69.5**
P170 Who was [E] created by?	870	**72.6**	54.1	57.7	56.8	70.9	71.6	72.3
P175 Who performed [E]?	1000	56.6	47.6	51.5	53.1	67.4	67.8	**68.6**
P176 Which company is [E] produced by?	1000	81.0	61.7	73.7	69.2	83.1	83.8	**84.2**
P19 Where was [E] born?	1000	75.3	25.4	41.8	52.9	80.7	81.9	**82.1**
P20 Where did [E] die?	1000	80.4	34.4	45.1	61.9	84.2	84.6	**85.1**
P26 Who is [E] married to?	1000	**89.7**	35.6	48.1	47.1	86.6	86.9	87.2
P264 What music label is [E] represented by?	1000	45.6	25.3	43.2	53.2	49.8	52.5	**55.7**
P276 Where is [E] located?	1000	84.9	74.9	77.3	77.1	84.2	85.1	**85.7**
P36 What is the capital of [E]?	886	**90.6**	77.3	78.9	91.7	89.7	90.1	90.5
P40 Who is [E]s child?	1000	85.0	19.2	33.8	39.7	87.1	88.2	**89.8**
P407 Which language was [E] written in?	646	86.2	77.1	82.5	81.9	88.5	89.1	**89.7**
P413 What is [E] famous for?	1000	74.3	75.7	71.5	53.8	83.2	84.9	**86.4**
P495 Which country was [E] created in?	1000	21.8	21.6	28.0	**34.8**	19.6	20.7	22.3
P50 Who is the author of [E]?	1000	73.0	75.7	77.8	77.2	78.3	79.6	**80.2**
P69 Where was [E] educated?	1000	73.1	26.4	41.8	38.6	74.1	**74.5**	**74.5**
P740 Where was [E] founded?	942	74.4	59.9	61.6	50.9	77.2	78.0	**79.0**
P800 What position does [E] play?	221	**74.7**	19.0	33.9	45.3	70.6	72.9	**74.7**
Macro-Average	-	72.0	49.7	56.7	59.9	74.8	75.7	**76.5**
Micro-Average	-	71.4	49.5	56.6	59.5	74.5	75.3	**76.2**

TABLE VI
ABLATION STUDY OF DISENTANGLEDQA ON THE DEVELOPMENT DATASET.

Methods	**Recall@10**	**QA Accuracy**
Full DisentangledQA	**55.9%**	**68.1**
w/o Topic Retriever	25.9% *(-30.0%)*	62.8 *(-5.3)*
w/o Attribute Retriever	51.6% *(-4.3%)*	63.3 *(-4.8)*
w/o Data Augmentation	52.8% *(-3.1%)*	64.6 *(-3.5)*
w/o Mask Mechanism	54.3% *(-1.6%)*	65.5 *(-2.6)*
w/o Strategy Predictor	55.9%	66.4 *(-1.7)*

TABLE VII
RECALL OF GOLDEN DOCUMENTS WITH DIFFERENT QUERY SETS. *CleanQ* DENOTES QUESTIONS WITHOUT STOP WORDS.

Recall	**All Found**				**At Least 1 Found**			
@N	3	5	8	10	3	5	8	10
CleanQ	37.3	39.7	39.7	39.7	56.8	60.7	60.7	60.7
NER	29.5	30.1	30.1	30.1	45.9	46.3	46.3	46.3
NGram	41.1	43.4	43.4	43.4	61.1	64.2	64.2	64.2
Noun	40.4	42.7	42.7	42.7	63.3	65.9	65.9	65.9
Our	**61.8**	**64.7**	**65.3**	**66.6**	**83.0**	**83.8**	**84.3**	**86.5**

It leverages the explicit topic information in the question and effectively filters a large amount or irrelevant context, with a high recall of true evidence for implicit QA. When attribute retriever is removed, the performance of evidence retrieval and QA accuracy decrease by 4.3% and 4.8, respectively. Moreover, it is observed that removing data augmentation in training the attribute retriever leads to a 3.1% performance drop in terms of Recall@10, indicating the importance of pseudo data in training a robust attribute-aware encoder. We disentangle the attribute information from the questions by employing a mask mechanism to ensure the implicit attributes are not disturbed by the explicitly mentioned topics. It is observed that removing the mask mechanism slightly affects paragraph-level recall by 1.6%, but significantly affects QA accuracy by 2.6. This observation shows that the mask mechanism is useful for the retriever to detect the true evidence from long semantic-related documents. Lastly, we remove the strategy predictor, resulting in a 1.7% QA performance drop, indicating that understanding the implicit reasoning strategy is helpful to answer inference.

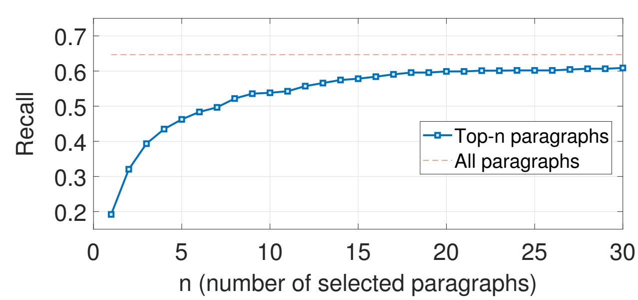
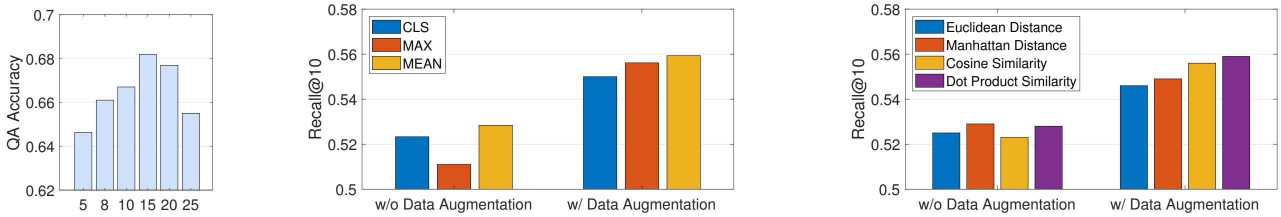


Fig. 7. The trade-off between the number of selected paragraphs and recall of golden paragraphs on the development set.

E. In-depth Analysis

The proposed method disentangles topic, attribute, and strategy from the implicit question to benefit retrieval and reasoning. We conduct an in-depth analysis of each component for answering implicit questions.



(a) Number of sentences (b) Pooling (c) Similarity

Fig. 8. Performance of attribute retriever. (a) QA accuracy of a different number of sentences, (b) Recall@10 of golden paragraphs with different pooling methods; and (c) Different vector similarity measures. *Aug.* denotes data augmentation and the dashed line denotes the best performance achieved without data augmentation.

1) Topic Retriever: We employ a multi-view query generator to retrieve documents which are related to the topics of the question. Table VII reports the recall of the required documents with different query sets and different numbers of the retrieved documents (i.e., $|\mathcal{D}_q|$). It is observed that *n-gram* and *nous* are more effective than *question* and *NER* as queries to retrieve the required documents. The multi-view query set achieves the best performance, indicating that it can effectively provide a more comprehensive query set and improve document-level retrieval performance. For the size of $\mathcal{D}_q$, recall increases with an increase in size, but the improvement is not significant when the size exceeds 5. Considering the balance between effect and efficiency, the size of $\mathcal{D}_q$ is set to 5. The recall of *all required documents* and *at least one required document* achieved by our topic retriever is 83.8% and 64.7%, respectively.

Then, a paragraph-level classifier based on RoBERTa-base model is trained to remove irrelevant paragraphs from $\mathcal{D}_q$. We set a threshold n to control the size of selected paragraphs for each question. Fig. 7 shows the recall with varying number of selected paragraphs in range 1 to 30. According to our statistics, $\mathcal{D}_q$ contains 155.8 paragraphs on average, and recall of golden paragraphs is 64.7% (i.e., the dashed line in Fig. 7). We generate $\mathcal{E}^{(t)_q}$ by selecting top-20 paragraphs for each question q, which reduces recall by 4.7% but removes 87.2% of the candidate paragraphs. In practice, the number of paragraphs to select can be dynamically controlled by adjusting n, so that proper number of paragraphs can be selected depending on the needs of recall and speed.

2) Attribute Retriever: We design the attribute retriever to select the top-k sentences to answer the implicit questions. We first compare the QA performance with a different number of selected sentences in $\mathcal{E}_q$. As shown in Fig. 8 (a), our method achieves the best performance when k is set to 15. The attribute retrieved is trained based on Sentence-Transform. We compare the performance with different pooling methods and different similarity functions. The pooling function has three optional strategies: 1) `CLS`: using the output of the `[CLS]` token, 2) `MEAN`: computing the mean of all the output vectors, and 3) `MAX`: computing a max-over-time of the output vectors.

Fig. 8 (b) indicates that `MEAN` is a more effective pooling method than `CLS` and `MAX`. Fig. 8 (c) shows that *dot-product* similarity achieves a slightly better performance than *cosine* similarity and is significantly better than *Euclidean* distance and *Manhattan* distance.

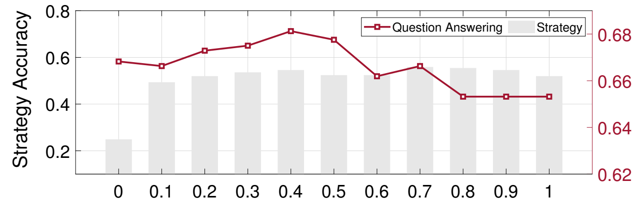


Fig. 9. Comparison with different λ in terms of the accuracy of strategy prediction and answer prediction.

In our experiment, we employ the `MEAN` pooling method and dot-product similarity to conduct dense retrieval. We also evaluate the performance of data augmentation as shown in Fig. 8 (b) and (c). It is observed that using pseudo examples as augmentation data significantly improves the effect of the attribute retriever.

3) Reasoning Strategy: In the training process, the combination parameter λ is used to control the contribution of strategy prediction and answer prediction. We vary λ in the range of [0,1] and plot the performance of strategy accuracy and QA accuracy in Fig. 9. It is observed that a too large λ slightly improves the strategy accuracy but affects the QA performance. Our method achieves the best QA performance when λ is set to 0.4. For the strategy prediction of five categories, the accuracy is 55.9% which shows that the latent strategy vector $\mathbf{h}^*$ defined in Eq (7) is representative of the implicit reasoning strategy.

F. Case Study

We conduct case study to better understand the proposed method. As shown in Fig. 10, we detail the outputs of our method for answering the question *"Did Football War last at least a month?"*. It is observed that the topic retriever is able to find the correct documents, i.e., *Football War* and *Month*. The attribute retriever can select sentences which are more related to *last at least* while *semantic retrieval* tends to select wrong sentences which contain the topic *Football War*. Our method correctly predicts that the reasoning strategy is *comparison*, which is helpful for answer inference. For the second question, it is necessary to retrieve the element set X required for the plant photosynthesis and the element set Y contained in the atmosphere of Mars, and judge that "*Does all the element in X present in Y?*".

Question 1: Did the Football War last at least a month?		**Answer**
BL	1. The **Football War**…colloquial: **Soccer War**…was a brief war fought between El Salvador and Honduras in 1969. 2. Although the nickname "**Football War**" implies that the conflict was due to a football match, the causes of the war go much deeper.	Yes (×)
Our	**Topic Retriever:** documents entitled [Football War, Month, Football, War, Football Football] **Attribute Retriever:** 1. Its **duration** is about **27.21222 days** on average. 2. The actual war had **lasted just over four days**, but it would…to arrive at a final peace settlement. **Strategy Predictor:** Comparison	No (√)
Question 2: Are all the elements plants need for photosynthesis present in atmosphere of Mars?		**Answer**
BL	1. Photosynthetic organisms…food directly from **carbon dioxide and water** using energy from light. 2. …CO2 is the main component of the **Martian atmosphere**.	No (×)
Our	**Topic Retriever**: documents entitled [Photosynthesis, Atmosphere of Mars, Atmosphere…] **Attribute Retriever:** 1. The **atmosphere of Mars consists of** 96% carbon dioxide, …along with traces of **oxygen and water.** 2. Total photosynthesis …include the amount of…, rate at which **carbon dioxide** can be supplied to the chloroplasts to support photosynthesis, the **availability of water**, and … **Strategy Predictor:** Entail	Yes (√)

Fig. 10. Case study of DisentangledQA. Golden documents are underlined. Topic-related words are marked in blue, and attribute-related words are marked in red. *semantic retrieval and reasoning* denotes the baseline method.

It was observed that the baseline method failed to retrieve evidence of Y, and our method successfully retrieved evidence containing X and Y and predicted the *entail* strategy, leading to a correct answer. These examples show that implicit question answering is challenging, because topics, attributes, and strategies are entangled in implicit questions. It is difficult to answer implicit questions by using the whole question directly. Our method provides richer and more accurate guidance information for each module by disentangling topic, attribute, and reasoning strategy from the question, thus improving the effectiveness.

V. Conclusion

In this paper, we propose DisentangledQA to answer implicit questions with an open-domain corpus. To better answer implicit questions, it disentangles the topic, attribute, and reasoning strategy from the questions to guide the retrieval and reasoning. The experiments on StrategyQA dataset show that the performance of DisentangledQA improved observably as a result of the underlying information of question and outperforms all the published models on the leaderboard. Moreover, the experiments on EntityQuestions dataset show that our method is effective to deal with general open-domain QA task. In the future, we would like to explore how to leverage linguistic knowledge to mine the required attributes in implicit questions, and how to exploit and encode latent reasoning strategies more accurately.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

[1] D. Chen and W. Yih, "Open-domain question answering," in *Proceedings of ACL: Tutorial Abstracts*, 2020, pp. 34–37.

[2] Y. Zhang, P. Nie, A. Ramamurthy, and L. Song, "Answering any-hop open-domain questions with iterative document reranking," in *Proceedings of SIGIR*, 2021, pp. 481–490.

[3] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *Proceedings of ACL*, 2017, pp. 1870–1879.

[4] Y. Nie, S. Wang, and M. Bansal, "Revealing the importance of semantic retrieval for machine reading at scale," in *Proceedings of EMNLP-IJCNLP*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., 2019, pp. 2553–2566.

[5] V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of EMNLP*, 2020, pp. 6769–6781.

[6] E. Cambria, L. Malandri, F. Mercorio, M. Mezzanzanica, and N. Nobani, "A survey on XAI and natural language explanations," *Information Processing and Management*, vol. 60, no. 103111, 2023.

[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *Proceedings of EMNLP*, 2016, pp. 2383–2392.

[8] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021.

[9] C. Sciavolino, Z. Zhong, J. Lee, and D. Chen, "Simple entity-centric questions challenge dense retrievers," in *Proceedings of EMNLP*, 2021, pp. 6138–6148.

[10] E. M. Voorhees, "The TREC-8 question answering track report," in *Proceedings of The Eighth Text REtrieval Conference, TREC*, ser. NIST Special Publication, vol. 500-246, 1999.

[11] J. Lee, M. J. Seo, H. Hajishirzi, and J. Kang, "Contextualized sparse representations for real-time open-domain question answering," in *Proceedings of ACL*, 2020, pp. 912–919.

[12] H. Cheng, Y. Shen, X. Liu, P. He, W. Chen, and J. Gao, "Unitedqa: A hybrid approach for open domain question answering," in *Proceedings of ACL/IJCNLP*, 2021, pp. 3080–3090.

[13] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," *Artificial Intelligence Review*, doi:10.1007/s10462-022-10248-8 2022.

[14] J. Wen, D. Jiang, G. Tu, C. Liu, and E. Cambria, "Dynamic interactive multiview memory network for emotion recognition in conversation," *Information Fusion*, vol. 91, pp. 123–133, 2023.

[15] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[16] S. E. Robertson and H. Zaragoza, "The probabilistic

relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[17] S. E. Robertson, S. Walker, and M. Hancock-Beaulieu, "Large test collection experiments on an operational, interactive system: Okapi at TREC," *Information Processing and Management*, vol. 31, no. 3, pp. 345–360, 1995.

[18] K. Lee, M. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," in *Proceedings of ACL*, 2019, pp. 6086–6096.

[19] W. Xiong, X. L. Li, S. Iyer, J. Du, P. S. H. Lewis, W. Y. Wang, Y. Mehdad, S. Yih, S. Riedel, D. Kiela, and B. Oguz, "Answering complex open-domain questions with multi-hop dense retrieval," in *Proceedings of ICLR*, 2021.

[20] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.

[21] Q. Liu, R. Mao, X. Geng, and E. Cambria, "Semantic matching in machine reading comprehension: An empirical study," *Information Processing and Management*, 2023.

[22] R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum, "Multi-step retriever-reader interaction for scalable open-domain question answering," in *Proceedings of ICLR*, 2019.

[23] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," in *Proceedings of ICLR*, 2020.

[24] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, "Generation-augmented retrieval for open-domain question answering," in *Proceedings of ACL/IJCNLP*, 2021, pp. 4089–4100.

[25] M. J. Seo, J. Lee, T. Kwiatkowski, A. P. Parikh, A. Farhadi, and H. Hajishirzi, "Real-time open-domain question answering with dense-sparse phrase index," in *Proceedings of ACL*, 2019, pp. 4430–4441.

[26] J. Lee, M. Sung, J. Kang, and D. Chen, "Learning dense representations of phrases at scale," in *Proceedings of ACL/IJCNLP*, 2021, pp. 6634–6647.

[27] V. Yadav, S. Bethard, and M. Surdeanu, "Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering," in *Proceedings of ACL*, 2020, pp. 4514–4525.

[28] T. Young, F. Xing, V. Pandelea, J. Ni, and E. Cambria, "Fusing task-oriented and open-domain dialogues in conversational agents," in *Proceedings of AAAI*, 2022, pp. 11 622–11 629.

[29] T. H. Alwaneen, A. M. Azmi, H. A. Aboalsamh, E. Cambria, and A. Hussain, "Arabic question answering system: A survey," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 207–253, 2022.

[30] J. Ni, V. Pandelea, T. Young, H. Zhou, and E. Cambria, "Hitkg: Towards goal-oriented conversations via multi-hierarchy learning," in *Proceedings of AAAI*, 2022, pp. 11 112–11 120.

[31] G. Tu, J. Wen, C. Liu, D. Jiang, and E. Cambria, "Context- and sentiment-aware networks for emotion recognition in conversation," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 699–708, 2022.

[32] w. Li, L. Zhu, and E. Cambria, "Taylor's theorem: A new perspective for neural tensor networks," *Knowledge-Based Systems*, vol. 228, no. 107258, 2021.

[33] H. Xu, H. Peng, H. Xie, E. Cambria, L. Zhou, and W. Zheng, "End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization," *World Wide Web*, vol. 23, pp. 1989–2002, 2020.

[34] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proceedings of ICCV 2017*, 2017, pp. 1839–1848.

[35] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 5947–5959, 2018.

[36] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[37] 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," *CoRR*, vol. abs/1907.11692, 2019.

[38] R. Mao and X. Li, "Bridging towers of multitask learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification," in *Proceedings of AAAI*, 2021, pp. 13 534–13 542.

[39] M. Ge, R. Mao, and E. Cambria, "Explainable metaphor identification inspired by conceptual metaphor theory," in *Proceedings of AAAI*, 2022, pp. 10 681–10 689.

[40] Y. Lai, C. Zhang, Y. Feng, Q. Huang, and D. Zhao, "Why machine reading comprehension models learn shortcuts?" in *Findings of ACL*, 2021, pp. 989–1002.

[41] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, "Multi-hop reading comprehension through question decomposition and rescoring," in *Proceedings of ACL*, 2019, pp. 6097–6109.

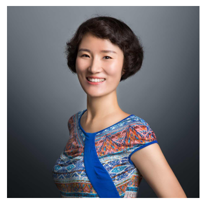
[42] E. Perez, P. S. H. Lewis, W. Yih, K. Cho, and D. Kiela, "Unsupervised question decomposition for question answering," in *Proceedings of EMNLP*, 2020, pp. 8864–8880.

[43] T. Wolfson, M. Geva, A. Gupta, Y. Goldberg, M. Gardner, D. Deutch, and J. Berant, "Break it down: A question understanding benchmark," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 183–198, 2020.

[44] 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 *Proceedings of ACL*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020, pp. 7871–

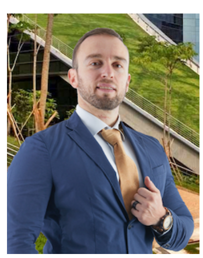
7880.
[45] L. Pan, W. Chen, W. Xiong, M. Kan, and W. Y. Wang, "Unsupervised multi-hop question answering by question generation," in *Proceedings of NAACL-HLT*, 2021, pp. 5866–5880.
[46] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic VQA: disentangling reasoning from vision and language understanding," in *Proceedings of NeurIPS*, 2018, pp. 1039–1050.
[47] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum, "CLEVRER: collision events for video representation and reasoning," in *Proceedings of ICLR*, 2020.
[48] Z. Chen, J. Mao, J. Wu, K. K. Wong, J. B. Tenenbaum, and C. Gan, "Grounding physical concepts of objects and events through dynamic visual reasoning," in *Proceedings of ICLR*, 2021.
[49] S. Min, V. Zhong, R. Socher, and C. Xiong, "Efficient and robust question answering from minimal context over documents," in *Proceedings of ACL*, 2018, pp. 1725–1735.
[50] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of EMNLP-IJCNLP*, 2019, pp. 3980–3990.
[51] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs," in *Proceedings of NAACL-HLT*, 2019, pp. 2368–2378.
[52] C. Clark, K. Lee, M. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "Boolq: Exploring the surprising difficulty of natural yes/no questions," in *Proceedings of NAACL-HLT*, 2019, pp. 2924–2936.
[53] V. Yadav, S. Bethard, and M. Surdeanu, "If you want to go far go together: Unsupervised joint candidate evidence retrieval for multi-hop question answering," in *Proceedings of NAACL-HLT*, 2021, pp. 4571–4581.
[54] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proceedings of ICML*, 2020, pp. 3929–3938.
[55] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019.
[56] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of ACL*, 2017, pp. 1601–1611.
[57] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of ACL*, 2013, pp. 1533–1544.
[58] P. Baudis and J. Sedivý, "Modeling of the question answering task in the yodaqa system," in *Proceedings of International Conference of the Cross-Language Evaluation Forum for European Languages*, ser. Lecture Notes in Computer Science, vol. 9283, pp. 222–228.

Qian Liu is a Postdoctoral Research Fellow in Nanyang Technological University, Singapore. She got Ph.D. degree in computer science from Beijing Institute of Technology (in 2020) and University of Technology Sydney (in 2022). Her research interests include natural language processing and information retrieval. She has published several papers in international conferences such as WWW, AAAI, COLING ect, and journals such as IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transaction on Knowledge and Data Engineering (TKDE), and IEEE Transaction on Fuzzy Systems (TFS).



Xiubo Geng is a Senior Applied Scientist in Microsoft STCA (Software Technology Center Asia). Her research interest includes machine learning, question answering, knowledge base, ranking, etc. She got her PhD degree in Institute of Computing Technology, Chinese Academy of Sciences. She has published a dozen of papers in top conferences including SIGIR, EMNLP, WWW, NIPS, IJCAI etc.

Yu Wang is a master student in Masters Program of Computer Science, University of Chicago. He received his bachelor's degree in computer science from Nankai University in 2020. His research interest covers Natural Language Processing, Deep Learning and Machine Learning. A couple of his papers has been published to top conferences and journals such as AAAI, ACL and ACM TIST.



Erik Cambria (F'22) is an Associate Professor at Nanyang Technological University, Singapore. He received the Ph.D. degree in computing science and mathematics through a joint programme between the University of Stirling, Stirling, U.K., and MIT Media Lab, Cambridge, MA, USA. His research focuses on the ensemble application of symbolic and sub-symbolic AI to natural language processing tasks such as sentiment analysis, dialogue systems, and financial forecasting. Erik is recipient of many awards, e.g., the 2019 IEEE Outstanding Early Career Award, he was listed among the 2018 AI's 10 to Watch, and was featured in Forbes as one of the 5 People Building Our AI Future. He is an IEEE Fellow, Associate Editor of many top-tier AI journals, e.g., INFFUS and IEEE TAFFC, and is involved in various international conferences as program chair and SPC member.

Daxin Jiang is Partner Chief Scientist of Microsoft STCA (Software Technology Center at Asia). Leads an R&D group with 140+ applied scientists and engineers to develop NLP technologies, applications and platforms, which support various Microsoft products, including Bing, Cortana, Teams, Outlook, XiaoICE, and Microsoft Cognitive Services. Years of experience of Research and Engineering in Machine Learning, Data Mining, Natural Language Processing, and Bioinformatics. Ph.D. in Computer Science from the Statue University of New York at Buffalo in 2005, Assistant Professor in the Computer Science and Engineering School of Nanyang Technological University, Singapore (2005-2006), and Lead Researcher in Microsoft Research Asia (2007-2011). Published 30+ papers with nearly 4000 citations. Won the SIGKDD Best Application Paper Award in 2008, and Runner-up for SIGKDD Best Application Paper Award in 2004.