A Survey of Extractive Question Answering

Luqi Wang
School of Information and Safety Engineering, Zhongnan University of Economics and Law
Wuhan, China
wangluqi2@stu.zuel.edu.cn

Liyin Qian
School of Information and Safety Engineering, Zhongnan University of Economics and Law
Wuhan, China
qianliyin@stu.zuel.edu.cn

Kaiwen Zheng
School of Information and Safety Engineering, Zhongnan University of Economics and Law
Wuhan, China
202112200042@stu.zuel.edu.cn

Sheng Li*
School of Information and Safety Engineering, Zhongnan University of Economics and Law
Wuhan, China
lisheng@zuel.edu.cn

2022 International Conference on High Performance Big Data and Intelligent Systems (HDIS) | 978-1-6654-9144-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/HDIS56859.2022.9991478

***Abstract*—Extractive question answering is one of the most important tasks in natural language processing (NLP) which has high research value. In order to sort out its development process, in this paper, it was divided into two categories roughly in this paper: single-span extractive question answering and multi-span extractive question answering, which are divided according to the number of spans. Studies on the two classifications are briefly described in the following context, following a certain logical relationship. And single-span extractive question is further divided into four categories to make a more detailed introduction. In addition, some works that can be done in the future is mentioned.**

***Index Terms*—question answering, attention mechanism, single-span extractive, multi-span extractive**

I. Introduction

Machine reading comprehension(MRC) means answering text-related questions based on a given text by a machine, which can measure machine comprehension of natural language. The reading comprehension task can be viewed as a supervised learning problem, that is, for a given certain amount of training examples, the goal is to learn a predictor that generates answers as output while the input consists of a piece of context and a natural language question. Among the various tasks of machine reading comprehension, Extractive Question Answering(EQA, also known as span prediction) is a popular section in it, which aims to extract a span or spans from the given context to answer the question. Due to the release of large-scale annotated datasets and well-designed neural models, rapid progress has been made in this area. Generally speaking, the EQA models usually consist of four components [1]: embeddings, feature extraction, context-question interaction, and answer prediction, as shown in Fig. 1.

- Embeddings: The input of the model(contexts and questions in natural language form)is encoded into a vector of fixed dimensions for subsequent processing by the computer.

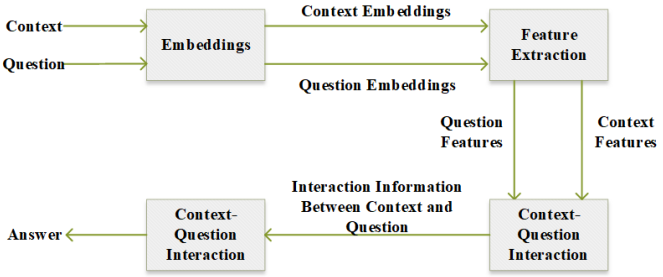

Fig. 1. The EQA process.

- Feature extraction: The word vector representations of articles and questions encoded by the embedding coding layer are received and processed to extract more contextual information. Commonly, Recurrent Neural Network(RNN), Convolutional Neural Network (CNN), and Transformer structure based on multi-head self-attentive mechanism are used in this module.
- Context-question interaction: This module often uses a unidirectional or bidirectional attention mechanism to emphasize the more important parts of the original text that are more associated to the question. At the same time, to dig deeper into the relationship between the article and the question, the interaction process between them may sometimes be executed several times, to simulate the human behavior of repeated reading when performing reading comprehension.
- Answer prediction: This module integrates the cumulative information obtained from the three above modules and makes the final answer prediction.

This paper describes the two different types of EQA: single- and multi-span extractive question answering in Section II and Section III. In Section IV, we summarize the future research directions.

* Corresponding author.

978-1-6654-9144-0/22/$31.00 ©2022 IEEE

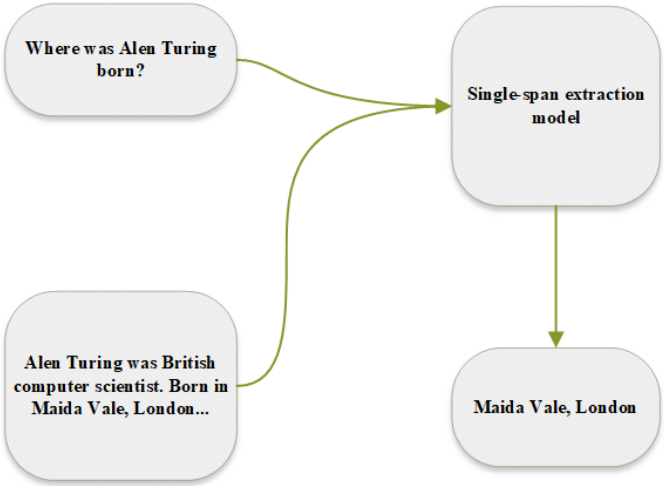

Fig. 2. An example of single-span EQA

II. Single-span Extractive QA

EQA can be further subdivided into two types, and the first one is called single-span EQA. As the name implies, for the given context and question, it aims to let the machine to extract only one text span from the given context as the correct answer, as shown in Fig. 2. That is, given the context symbolically represented as $C = \{t_1, t_2, ..., t_n\}$, and the question Q,this mission needs to extract a continuous sequence $S = \{t_i, t_{i+1}, ..., t_{i+k}\,(1 \leq)\, i \leq i + k \leq n\}$ as the right answer to question Q from context C, and this process is usually learned from the function F, so it can be represented as $S = F\,(C, Q)$ finally.

In this part, we divide single-span QA into four categories for a comprehensive understanding, also, we briefly describe several common datasets of it.

A. Extractive QA Using Attention Mechanism

Attention is a mechanism used to improve the effectiveness of encoder-decoder models based on Recurrent Neural Network(RNN), Long Short-Term Memory(LSTM), or Gated Recurrent Unit(GRU), and it can equip model with discrimination capability, allowing for increased flexibility in model learning. Specifically, in the EQA task, by providing different weights for sentences and words in the context.

Inspired by end-to-end neural architectures, [2] proposed a new end-to-end neural architecture based on match-LSTM(a model for textual entailment) [3] and Pointer Net(a sequence-to-sequence model) [4]. Specifically, they adopt a match-LSTM to predict whether the two components of textual entailment, a premise, and a hypothesis, have the inclusion relationship, that is, whether the premise includes the hypothesis. In position i of the hypothesis, the attention mechanism gets a weighted vector α_i representation of the premise, as shown in (1), (2).

$$\vec{\mathbf{G}}_i = tanh(\mathbf{W}^q\mathbf{H}^q + \mathbf{W}^r\vec{\mathbf{h}}^r_{i-1} + \mathbf{b}^p \otimes \mathbf{e}_Q) \tag{1}$$

$$\vec{\alpha}_i = softmax(\mathbf{w}^\top\vec{\mathbf{G}}_i + b \otimes \mathbf{e}_Q) \tag{2}$$

Where $\mathbf{W}^q$, $\mathbf{W}^p$, $\mathbf{W}^r$, $\mathbf{b}^p$, $\mathbf{w}$, b are parameters need to be learned, $\vec{\mathbf{h}}^r_{i-1}$ is the hidden vector of match-LSTM at position $i-1$. And the outer product $(\cdot \otimes \mathbf{e}_Q)$ produces a matrix or row vector by repeating the vector or scalar on the left for Q times. Then they further adapt the Pointer Net(Ptr-Net) to make the tokens of an output sequence source from the input sequence. Ptr-Net thinks of the attention mechanism as a pointer and then selects a position as an output symbol from the input sequence. With this model, they achieved a high exact match score on the dataset.

Reference [5] follows the work of [2] and presented the gated self-matching networks, also an end-to-end neural network model. Similar to [2], they use Ptr-Ne to position the answers. The difference from the above is that, first, they add an extra gate to the networks and name the network as gated attention-based network to the input of RNN, and assigns different weights to different parts of the article, rather than encoding them together:

$$g_t = sigmoid\,(W_g\,[u^p_t, c_t]) \tag{3}$$

$$\left[u^p_t, c^t\right]^* = g_t \odot \left[u^p_t, c^t\right] \tag{4}$$

where W_g is the parameters to be learned, u^p_t means passage representation and c_t is an attention-pooling vector of the whole question.Second, they presented a self-matching mechanism to gather valid information from the context and then predict answers. The combination of these two parts enriches the representation of the context and facilitates the subsequent prediction of answers.

Compared with the above two models, BIDAF [6] adopts a memory-less attention mechanism, that is, in the iterative process, the computation of attention for each time step is irrelevant to the attention of the previous moment and depends only on the query and the context paragraph of the current moment. Instead of encoding the contexts into fixed-size vectors, the attention of each time step and its attended vector are calculated based on the representations of the previous layer. First obtain a similarity matrix between H(output matrix of context) and U((output matrix of query):

$$\mathbf{S}_{tj} = \alpha\,(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \tag{5}$$

where $\mathbf{S}_{tj}$ denotes the similarity between the t-th context word and the j-th query word and is a scalar function. And then, they compute attention information not only in the query-to-context(Q2C) direction but also in the context-to-query (C2Q) direction where using S as a shared similarity matrix.

The above models are designed based on RNNs, but due to the sequential characteristic of RNNs, these models usually spend a lot of time on training and inference. To overcome this shortcoming, [7] proposed a new network named QANet, which discards the RNN, and its encoder consists of convolution and self-attention entirely for different function, where these two parts are respectively simulates local interactions and simulates global interactions. In this way, the model is trained 3 to 13 times faster and inferred 4 to 9 times faster on the same dataset.

In general, early studies used attention mechanisms to capture the rich interaction information between contexts and queries, enabling EQA to continue to evolve.

B. Unsurprived Extractive QA

In the following study, numerous methods add Pre-trained Language Models(PLMs), such as BERT [8], RoBERTa [9], and SpanBERT [10] to the pointer header to get the scope of the answer span, the wealth of information contained in PLMs can be effective assistance for reasoning. But these models and their fine-tuning models rely heavily on the laborious and time-consuming process of data annotation, if there is a lack of sufficient labeled data, it is difficult to train an efficient model. So it is another task to find approaches to use PLMs for EQA to maintain high performance in the data-less learning scenarios.

An effective solution is to use unsupervised learning, which means there is no setting about the question, context and answer data in the whole process, in other words, no labeled data.

Reference [11] has done seminal work on this task. In case there is no data, they generate context, question and answer triples S in an unsupervised way and then use these triples to synthesize the training data.

$$S = \{(c_i, q_i, a_i)\}_{i=1}^{N} \tag{6}$$

First, they sample a context from a target domain such as DBpedia. Then, they sample candidate answers relevant to the context and transfer them intofill-in-the-blank cloze questions. Finally, they use an unsupervised translator to convert cloze questions into natural questions. Specifically, they first use this data to generate a generative model via the unsupervised method and then use it as a training data generator to train a discriminative model Through this study, we can find that it is possible to finish EQA tasks without labeled data.

However, this method exists many overlapping terms between the generated questions and the context, which causes the model to favor word matching to predict the answer. Besides, limited by named entities and noun phrases, answer types will not be too rich, which has a limitation on model coverage [12]. To solve this problem, they introduce two solutions to increase the quality of triples. First, they create a dataset named REFQA, which focuses on questions with grammatical or lexical ambiguity from Wikipedia by using the referred documents, which can form context-question-answer triples with more information. Second, they iterate by REFQA dataset over the QA model to get more accurate answers. Specifically, they sample REFQA $S_I = \{(c_i, q_i, a_i)\}_{i=1}^{N}$ to train a model $P(a|c, q)$ based on BERT by maximizing:

$$\sum_{S_I} \log P\left(a_i|c_i, q_i\right) \tag{7}$$

And then they refine the question-answer pairs using the resulting QA model to remove noisy examples and repeat the training procedure.

Besides, the method mentioned in [11] also has problem that the resulting improvement is not significant when compared to simpler problem generators [13]. So based on the [11], they propose a simpler and more intuitive approach to problem generation, and the difference is that this approach is retrieval and template-based. That is, to generate a question, retrieve a sentence from the corpus relevant to the present context. After obtaining all questions for all (context, answer) pairs, using these data to fine-tune a pre-trained BERT model and gain the final model.

In general, the focus of unsupervised methods is to create an effective corpus that can be used to train a supervised model on it.

C. Few-shot Extractive QA

Unsupervised extractive QA is suitable for situations where there is no data at all, and this is an extreme case. In certain cases, few samples can be used, which can also be called few-shot learning. Under the condition of few samples, the standard models show bad performance, the reason for this is the inconsistency between pre-training goals and fine-tuning goals [14]. And although the fine-tuning PLMs perform well in EQA tasks, these standard fine-tuning settings may cause over-fitting in few-shot settings [15]. Therefore, studies on this subject have emerged one after another.

The above-mentioned reference [11] and [12] perform few-shot settings in the experiment, demonstrating the feasibility of the study. To overcome the situation of using pre-trained models fine-tuning with few samples but getting bad results, [14] introduced a new pre-training method called recurring span selection, for a context with sets of recurring spans(it can also be understood as repeating words and phrases), they use this self-supervised method to produce question-answer pairs from unlabeled context and then use these data to pre-training. The model will choose the correct one according to the probability and mask the remaining spans:

$$P\left(s = i|C, q\right) = \frac{\exp\left(\mathbf{x}_i^T \mathbf{S} \mathbf{x}_q\right)}{\sum_j \exp\left(\mathbf{x}_j^T \mathbf{S} \mathbf{x}_q\right)} \tag{8}$$

$$P\left(e = i|C, q\right) = \frac{\exp\left(\mathbf{x}_i^T \mathbf{E} \mathbf{x}_q\right)}{\sum_j \exp\left(\mathbf{x}_j^T \mathbf{E} \mathbf{x}_q\right)} \tag{9}$$

where s represents the start position and e represents end position of the answer span, $\mathbf{x}_i$ represents a contextualized token representation and $\mathbf{S}$, $\mathbf{E}$ are parameter matrices. Masked spans will be replaced by a special token $[QUESTION]$, which will be later used in fine-tuning to select the answer span. By training on this new model, it can achieve high accuracy even with a small number of samples.

Although [11] solves the problem of the bad result on standard pre-training models, it spends a lot of time on computing resources in the pre-training process [15]. So they use prompt-tuning to few-shot settings without any extra pre-training steps. It can be simply understood as adding supplementary text to the input of the task to make better use of the pre-trained model. Processing of input text information according to specific templates, to adapt and take full advantage of the forms processed by the pre-trained language model, allowing

TABLE I
AN EXAMPLE OF CONSTRUCTED CLOZE.

Passage(P)	Question(Q)	Answer(A)
...Tancred was instrumental in the conquest of Jerusalem and...	What major conquest did Tancred play a role in?	Jerusalem

various downstream tasks to "accommodate" the pre-trained model. Specifically, the $[MASK]$ tokens will be added to task-specific prompt templates, and then the Masked Language Modeling(MLM) head generates the masked positions to follow-up predictions.

Few-shot learning is a machine learning scenario, the above article describes several ways to solve the problem and semi-supervised learning also can be a solution to it.

Reference [16] presented a system that needs to provide the base documents and a small set of question-answer pairs, as shown in Table I, and then produces cloze-style questions from unlabeled contexts. The generated questions are used to pre-train and the question-answer pairs are used to fine-tune.

In general, there are some commonalities between unsupervised and few-shot learning. And unsupervised methods focus on generating data while few-shot learning methods focus on the pre-training process.

D. Other Models and Studies

In addition to the several common categories mentioned above, the use of other methods for EQA has also been actively studied by scholars.

1) Model research: Reference [17] proposed a new pre-training model named UNIfied pre-trained Language Model (UNILM). To pre-train the model, it uses three different types of language modeling tasks, they are unidirectional, bidirectional, and sequence-to-sequence prediction. Reference [11] designed a new pre-training task for BERT language models in the form of cloze-like question and answer tasks, and it trains language models in deep semantic understanding, rather than a mnemonic way of encoding specific knowledge in the model parameters, thus improving the performance of downstream question and answer tasks. Reference [18] created a new model named StructBERT extended from BERT. By using two new linearization strategies, it introduces a new contextual representation type, making the BERT model pre-train with language structures. And to obtain richer semantic information, [12] and [15] use the external knowledge base(KB) such as Wikipedia as information resources. Inspired by contextual bandit learning, [19] uses user feedback with few-shot settings to improve the system on-the-fly.

2) Ancillary research: That is, provide resources (not discussing official resources) or reference help for EQA research. As mentioned earlier, [12] create a dataset named REFQA, which focuses on questions with grammatical or lexical ambiguity from Wikipedia by using the referred documents. Reference [11] proposed a new benchmark using multi-way aligned. It covers seven languages including English, German, Spanish, Hindi, Arabic, Vietnamese and Simplified Chinese, making up for the scarcity of languages other than English. Reference [20] studied the difference between distinct pre-trained language models based on transformer using datasets of different levels of difficulty.

E. Benchmarks

To test various EQA models, several benchmarks have been published one after another, and these datasets are used in the previously mentioned literature, which we present using Table II for a more visual presentation.

III. MULTI-SPAN EXTRACTIVE QA

The previous section details the various EQA methods that have been used this year, but what these methods have in common is that there is only one answer, which means that they can only answer single-answer questions. However, in practical applications, the answers to the questions often involve multiple types and multiple text strings, or the answers need to be reasoned, the performance of the previous model will be significantly degraded in these scenes, as shown in Fig. 3. Therefore, a trend in extractive QA is to tackle multi-span tasks.

A. Benchmarks

Limited by the lack of data sets, few studies have been conducted on multi-span tasks. In 2019, once the DROP [21] dataset was released, studies on multi-span extraction emerged one after another. DROP contains 96.6K question-answer pairs and the passages are extracted from Wikipedia. The answers to these questions cover a variety of types, such as time, dates, numbers, etc. Also, some questions have multiple answers or require some reasoning to answer, such as numerical comparison. In the same year, Allen Institute for Artificial Intelligence [22] released a crowdsourced dataset named Quoerf, which contains over 24K span-selection questions related to passages from Wikipedia, and it can resolve coreference among entities from different articles. Fig. 4 shows some instances in these two datasets.

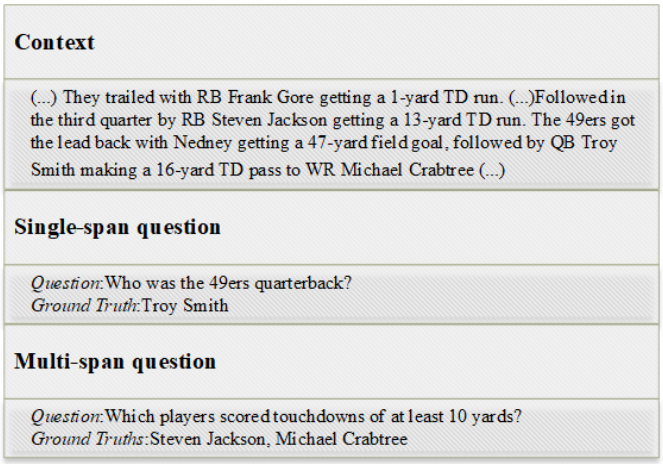

Fig. 3. Compairation of single-span and multi-span EQA

TABLE II
BENCHMARKS FOR SINGLE-SPAN EQA.

Name	Number of questions	Number of documents	Data resource	Time	Features
SQuAD 1.0	108k	23k paragraphs from 536 articles	Wikipedia	2016	Contains 100,000 (question, text, answer) triples, and the answer of each question comes from a paragraph of text in related article.
SQuAD 2.0	150k	23k paragraphs from 536 articles	Wikipedia	2018	50,000 new human-written questions have been added to SQuAD 1.0, and the questions do not necessarily have corresponding answers.
NewsQA	120k	13k	CNN	2016	The data is presented in question-answer pairs, with questions from more than 10,000 CNN p.ress releases and answers from corresponding articles and text passages.
TriviaQA	40k	660k	Wikipedia	2017	The questions in the dataset are complex, while there is a lot of syntactic or lexical variation in the questions and the corresponding answer sentences
HotpotQA	113k	113k	Wikipedia	2018	A multi-hop reasoning quiz dataset.The training set was divided into three difficulty levels: easy, medium, and hard, with medium accounting for the major part.
NQ	over 300k	the passage of the question is a website from Wikipedia	Wikipedia	2019	It consists of four tuples (question, Wikipedia page, long answer, short answer) , the questions in it are from the queries that real users have asked.

B. Methods

At the same time as publishing the dataset, [21] proposed a numerically-aware QANet model based on QANet, which can deal with three new types of answers, they are, counts, question spans, Number addition, and subtraction. Based on QANet, NAQANet adds four output layers called heads to handle different problem types. These four heads include (1) Passage span, designed to handle the type of single-span questions that are involved in SQuAD. The probabilities of the starting and ending positions from the context can be calculated as:

$$\mathbf{p}^{c\_start} = softmax\left(FFN\left[\mathbf{M}_0; \mathbf{M}_1\right]\right) \tag{10}$$

$$\mathbf{p}^{c\_end} = softmax\left(FFN\left[\mathbf{M}_0; \mathbf{M}_2\right]\right) \tag{11}$$

where $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_2$ are the outputs of the QANet encoder and FFN is a feed-forward network with two layer. (2) Question span, designed to handle the questions that the answer is in the question itself rather than passages. (3) Count, designed to handle questions that need to be counted. (4) Arithmetic expression, designed to handle questions requiring numerical operations. To decide which head needs to be chosen while answering a question, they train a 4-way categorical variable called head predictor. Reference [23] then released an improved version of NAQANet, NABERT+. They first changed the QANET encoder into BERT. The second enhancement was that they added predefined standard numbers into the arithmetic whether or not they appear in the passage.

Although NAQANet and NABERT+ have made some progress compared to previous single-span methods, Multi-Type Multi-Span Network(MTMSN) [24] is the first real solution for multi-span questions. The model contains two parts. To determine the number of spans while extracting, a dedicated categorical variable was trained in the first part. To search for the most probable non-overlapping span set, the second part promoted the single-span head method for extracting spans using the non-maximum suppression (NMS) algorithm. The categorical variable determined how many spans needed to be extracted. Besides, MTMSN added two components that were not related to span. One for handling logical negation, such as "How many percent were not American?". The other is for enhancing the arithmetic head.

While MTMSN trying to generalize the available method to tackle multi-span questions, [25] directly proposed a new method to solve the problem, that is, transforming them into sequence tagging tasks. Specifically, they first tag input text sequences using the BIO format where B marks the beginning of the answer span, I marks the inside of the span, and O marks the token that is not part of the answer fragment. Then they use multi-task learning with a multi-output head structure. At last, they use Beam Search to select the top-k predictions as final answers and remove the wrong sequences. The probabilities of the starting and ending positions from the context can be calculated as:

$$\mathbf{p}^{c\_start} = softmax\left(\mathbf{W}^S \mathbf{T}^C\right) \tag{12}$$

$$\mathbf{p}^{c\_end} = softmax\left(\mathbf{W}^E \mathbf{T}^C\right) \tag{13}$$

where $\mathbf{W}^S$, $\mathbf{W}^E$ are learned vectors and $\mathbf{T}$ is BERTs output on a (C, Q) input.

Different from the above models, reference [26] takes a different approach, they uses different single-span models to find fragments of multiple answers. Then, they rewrite the fragments to get complete answers. It is the first method to deal with multi-span questions using single-span supervision.

Besides, sometimes we need to carry out some reasonings or calculations using multi answers, here, we treat it as special

multi-span tasks. Although the NAQANet has made some attempts in numerical questions, it is still unable to do numerical reasoning. To tackle this shortcoming, [27] created a novel model named NumNet which consists of encoding module, reasoning module and prediction module. They consider the key point of numerical reasoning is numerical comparison. So they encode the numerical relationship between the questions and the numbers into a graph as its topology, and use a numerically-aware graph neural network to do reasoning. Similarly, [28] designed a heterogeneous directed graph with nodes consisting of entities and numbers of different types, and edges showing the relationships between the different types. The models mentioned above rarely attach importance to explicitly modeling of evidence, which is usually the critical clue from question to answer. Thus, [29] performed reasoning over a multi-grained evidence graph.

IV. The Future of EQA

Although many methods and models have been proposed to tackle EQA tasks, with the increase of various demands, there are still some things to do in future works. We have classified them simply in this section.

A. Datasets

As we can see from the above contents, the lack of datasets can be an obstruction for EQA. For single-span tasks, the datasets are adequate but contain some problems in these datasets. As some questions are unable to answer, or require more complex cross-temporal or documentary reasoning, or need to process longer documents. For multi-span tasks, the biggest problem is that there are not enough datasets for us to do the study. In the future, if there are more complex and more diverse datasets are created, research in this area can turn to the next level. Reference [15] inspires us that knowledge graphs can be helpful external Information which are independent of the datasets.

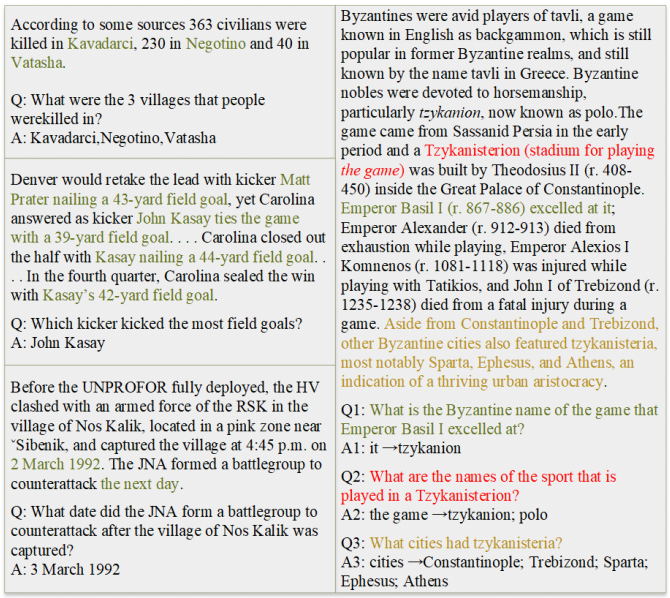

Fig. 4. Examples of the datasets (The left side is DROP, the right side is Quoerf.)

B. Models

Reference [7] mentions that both training and inference processes are time-consuming, and it is impossible to encode a very long document with the network. Solutions such as transformer and non-recurrent models have been mentioned above. But how to balance the data scale and time consumption is a topic worth studying. Reference [29] points out the previous methods do not provide evidence to support the result, which means the interpretability of the models is not enough. Actually, in the current systems, interpretability is always ignored, they can show the result to us but cannot provide reasonable explanations, and this is helpless to following tasks. The systems in the future should be able to provide not only the final answer, but also the rationale behind the predictions, so that users can decide whether they can trust the outputs and make use of them.

V. Conclusion

In this paper, we give a general description and summary of the EQA methods and models in recent years. We roughly divide the models into three categories, and the first two parts are highlighted. It can be seen that there exists a large amount of research on EQA in the field, covering various aspects such as dataset collection, model creation and improvement. In the future, we will continue to study in depth the improvement of EQA models to enrich our research.

References

[1] D. Chen, *Neural reading comprehension and beyond.* Stanford University, 2018.

[2] S. Wang and J. Jiang, "Machine comprehension using match-lstm and answer pointer," *arXiv preprint arXiv:1608.07905*, 2016.

[3] S. Wang and J. Jiang, "Learning natural language inference with LSTM," *CoRR*, vol. abs/1512.08849, 2015.

[4] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[5] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 189–198.

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

[7] 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," *arXiv preprint arXiv:1804.09541*, 2018.

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

[9] 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," *arXiv preprint arXiv:1907.11692*, 2019.

[10] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[11] P. Lewis, L. Denoyer, and S. Riedel, "Unsupervised question answering by cloze translation," *arXiv preprint arXiv:1906.04980*, 2019.

[12] Z. Li, W. Wang, L. Dong, F. Wei, and K. Xu, "Harvesting and refining question-answer pairs for unsupervised qa," *arXiv preprint arXiv:2005.02925*, 2020.
[13] A. R. Fabbri, P. Ng, Z. Wang, R. Nallapati, and B. Xiang, "Template-based question generation from retrieved sentences for improved unsupervised question answering," *arXiv preprint arXiv:2004.11892*, 2020.
[14] O. Ram, Y. Kirstain, J. Berant, A. Globerson, and O. Levy, "Few-shot question answering by pretraining span selection," *arXiv preprint arXiv:2101.00438*, 2021.
[15] J. Wang, C. Wang, M. Qiu, Q. Shi, H. Wang, J. Huang, and M. Gao, "Kecp: Knowledge enhanced contrastive prompting for few-shot extractive question answering," *arXiv preprint arXiv:2205.03071*, 2022.
[16] B. Dhingra, D. Pruthi, and D. Rajagopal, "Simple and effective semi-supervised question answering," *arXiv preprint arXiv:1804.00720*, 2018.
[17] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[18] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and L. Si, "Structbert: Incorporating language structures into pre-training for deep language understanding," *arXiv preprint arXiv:1908.04577*, 2019.
[19] G. Gao, E. Choi, and Y. Artzi, "Simulating bandit learning from user feedback for extractive question answering," *arXiv preprint arXiv:2203.10079*, 2022.
[20] K. Pearce, T. Zhan, A. Komanduri, and J. Zhan, "A comparative study of transformer-based language models on extractive question answering," *arXiv preprint arXiv:2110.03142*, 2021.
[21] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs," *arXiv preprint arXiv:1903.00161*, 2019.
[22] Y. Zhou, J. Bao, C. Duan, H. Sun, J. Liang, Y. Wang, J. Zhao, Y. Wu, X. He, and T. Zhao, "Opera: Operation-pivoted discrete reasoning over text," *arXiv preprint arXiv:2204.14166*, 2022.
[23] J. Kinley and R. Lin, "Nabert+: Improving numerical reasoning in reading comprehension," *URL https://github. com/raylin1000/drop-bert*, 2019.
[24] M. Hu, Y. Peng, Z. Huang, and D. Li, "A multi-type multi-span network for reading comprehension that requires discrete reasoning," *arXiv preprint arXiv:1908.05514*, 2019.
[25] A. Efrat, E. Segal, and M. Shoham, "Tag-based multi-span extraction in reading comprehension," *arXiv*, pp. arXiv–1909, 2019.
[26] T. Takahashi, M. Taniguchi, T. Taniguchi, and T. Ohkuma, "Multi-span extractive reading comprehension without multi-span supervision," in *European Conference on Information Retrieval*. Springer, 2021, pp. 401–409.
[27] Q. Ran, Y. Lin, P. Li, J. Zhou, and Z. Liu, "Numnet: Machine reading comprehension with numerical reasoning," *arXiv preprint arXiv:1910.06701*, 2019.
[28] K. Chen, W. Xu, X. Cheng, Z. Xiaochuan, Y. Zhang, L. Song, T. Wang, Y. Qi, and W. Chu, "Question directed graph attention network for numerical reasoning over text," *arXiv preprint arXiv:2009.07448*, 2020.
[29] Y. Zhou, J. Bao, H. Sun, J. Liang, Y. Wu, X. He, B. Zhou, and T. Zhao, "Evidr: Evidence-emphasized discrete reasoning for reasoning machine reading comprehension," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2021, pp. 439–452.