Learning to Attend On Essential Terms: An Enhanced Retriever-Reader Model for Scientific Question Answering

Jianmo Ni[1][*], **Chenguang Zhu**[2], **Weizhu Chen**[2], **Julian McAuley**[1]
[1] Department of Computer Science, UC San Diego
[2] Microsoft AI+R
{jin018,jmcauley}@ucsd.edu, {chezhu,wzchen}@microsoft.com

Abstract

Scientific Question Answering (SQA) is a challenging open-domain task which requires the capability to understand questions and choices, collect useful information, and reason over evidence. Previous work typically formulates this task as a reading comprehension or entailment problem given evidence retrieved from search engines. However, existing techniques struggle to retrieve indirectly related evidence when no directly related evidence is provided, especially for complex questions where it is hard to parse precisely what the question asks. In this paper we propose a retriever-reader model that learns to attend on essential terms during the question answering process. We build 1) an essential-term-aware 'retriever' which first identifies the most important words in a question, then reformulates the queries and searches for related evidence 2) an enhanced 'reader' to distinguish between essential terms and distracting words to predict the answer. We experimentally evaluate our model on the ARC dataset where it outperforms the existing state-of-the-art model by 8.1%.

1 Introduction

Recent advances in question answering systems have shown great effectiveness in a wide range of tasks such as machine reading comprehension (MRC) (Rajpurkar et al., 2016, 2018), community question answering (CQA) (Nakov et al., 2016), visual question answering (VQA) (Antol et al., 2015), among others. However, the performance of modern systems drops significantly when the need arises to understand long and complex questions, and reason over multiple passages, rather than simple text matching. Moreover, QA tasks become inherently more difficult when dealing with questions for which there is little evidence available (e.g. open-domain setting). As a result, it is essential to incorporate commonsense knowledge or improve retrieval capability to make sure related knowledge are collected (Chen et al., 2017).

Scientific Question Answering (SQA) is a representative task that exhibits the above-mentioned challenges. In this paper, we study question answering on the AI2 Reasoning Challenge (ARC) science QA dataset (Clark et al., 2018). This dataset contains elementary-level multiple-choice science questions from standardized tests and a large corpus of relevant information gathered from search engines. The dataset is partitioned into 'Challenge' and 'Easy' sets. The challenge set consists of questions that cannot be answered correctly by any of the Point-wise Mutual Information (PMI) or Information Retrieval (IR) based solvers. Most of the baseline models achieve only slightly better (or sometimes worse) performance than random guessing, which demonstrates that the existing models are not strong enough or otherwise unsuited for this kind of QA task.

To address these difficulties, we propose an essential-term-aware Retriever-Reader (ET-RR) model that learns to attend on essential terms during retrieval and reading. Specifically, we develop a two-stage method with an essential-term-aware retriever, followed by an attention-enhanced reader. Our model achieves a score of 43.9 on the *Dev* set and 36.61 on the *Test* set which significantly beats the previous SOTA.

[*]Work in progress. Most work done during internship at Microsoft, Redmond.

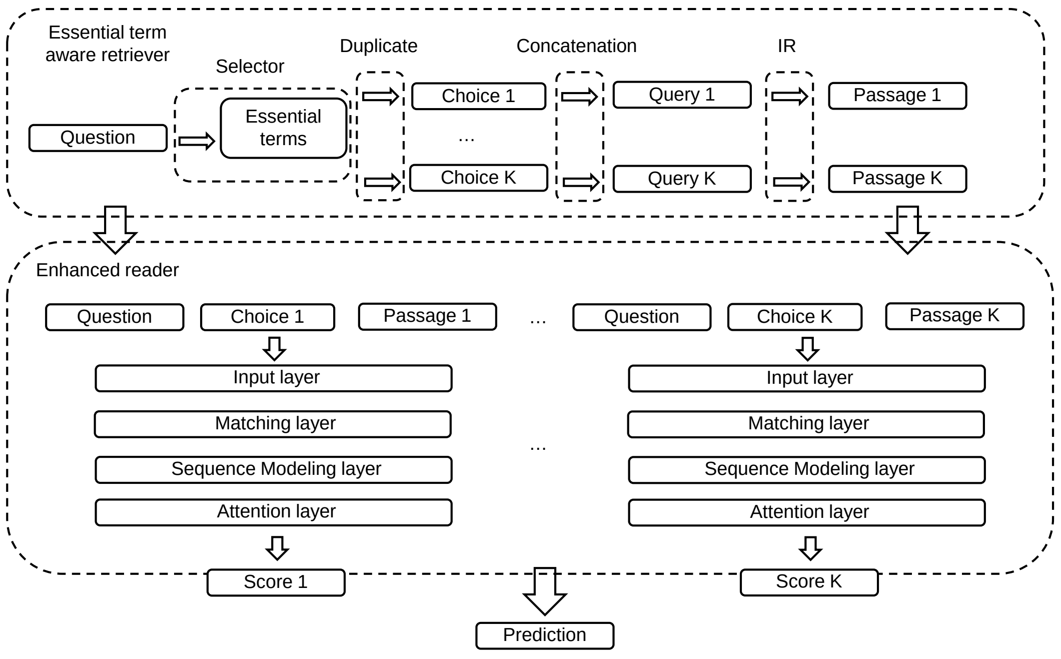

Figure 1: Model structure for essential-term-aware retriever-reader model.

2 Related Work

Open domain QA has been extensively studied in recent years. Recent efforts have followed the search-and-answer strategy and achieved strong performance (Chen et al., 2017). While many methods along these lines benefit from enhanced 'readers,' there has recently been growing interest in building better retrievers. (Wang et al., 2018) proposed a Reinforced Ranker-Reader model that ranks the retrieved evidence and assign them different weight before processing by the reader. However, few of these works have focused on searching other than ranking.

There has been a line of work studying science QA ranging from question understanding (Khashabi et al., 2017), information aggregation (Kwon et al., 2018) and multi-hop inference (Jansen et al., 2018). (Khashabi et al., 2017) worked on the problem of finding essential terms in a question, which has the same motivation as our approach. They handcrafted 100+ features and proposed an SVM classifier to uncover essential terms within a question. They also published a dataset containing over 2,200 science questions annotated with essential terms. In this paper, we leverage this dataset and build an essential term selector by training on it.

More recently, (Boratko et al., 2018) developed a labeling interface to obtain high quality labels for the ARC dataset. One interesting finding is that human annotators tend to retrieve better evidence after they reformulate the search queries which are originally constructed by a simple concatenation of question and choice; by feeding the evidence from obtained by human-reformulated queries into a pre-trained MRC model and achieved 54.7% accuracy on a subset of 47 questions. This shows the potential for a 'human-like' retriever to boost the performance on this task. Inspired by this work, we focus on picking essential terms to reformulate more efficient queries, similar to those that a human would construct.

3 Approach

In this section, we introduce the proposed essential-term-aware retriever-reader model (ET-RR). As shown in fig. 1, we build a term selector to figure out which terms are essential in a question. The selected terms are used to formulate a more efficient query that enables the retriever to obtain related evidence. The retrieved evidence is then fed to the reader to predict the final answer.

We first present the detail of the proposed retriever-reader model (ET-RR), then we demonstrate the selector module (ET-Net) within the retriever.

3.1 Proposed Retriever-Reader model

For each question $\mathbf{Q} \in \mathbb{R}^q$ along with its K choices $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\} \in \mathbb{R}^{K \times c}$, the selector chooses a subset of essential terms $\mathbf{E}$ from $\mathbf{Q}$, which are then concatenated with each $\mathbf{C}_k$ to formulate a query. The queries for each choice are sent to the retriever (e.g. Elastic Search) and K passages of evidence $\mathbf{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_K\} \in \mathbb{R}^{K \times p}$ are returned. Given these text sequences $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{P}$, the reader will determine the score for each triple $\{\mathbf{Q}, \mathbf{C}_k, \mathbf{P}_k\}$, which are then ranked to obtain the final answer. The details of the reader are described as follows; to simplify notation, we ignore the subscript k until the final output layer.

3.1.1 Input Layer

In this layer, all text inputs—the question, choices, and passages (i.e., retrieved evidence) are converted into embedded representations. Similar to Wang (2018), we consider the following components for each word:

- Word Embedding. Pre-trained GloVe word embedding with dimensionality $d_w = 300$.
- Part-of-Speech Embedding and Named-Entity Embedding. The part-of-speech tags and named entities for each word are mapped to embeddings with dimension 16.
- Relation Embedding. A relation between each word in $\mathbf{P}$ and any word in $\mathbf{Q}$ or $\mathbf{C}$ is mapped to an embedding with dimension size 10. The relation is obtained by querying ConceptNet (Speer et al., 2017).
- Feature Embeddings. Three handcrafted features are considered to enhance the word representations:
  - Word Match. If each word or its lemma of $\mathbf{P}$ exists in $\mathbf{Q}$ or $\mathbf{C}$, then this feature is 1 (0 otherwise).
  - Word Frequency. A logarithmic term frequency is also calculated for each word.
  - Essential Term. This feature is 1 if the word is an essential term (0 otherwise).

For $\mathbf{Q}, \mathbf{C}, \mathbf{P}$, all these components are concatenated together to obtain the final word representation $\mathbf{w}_Q \in \mathbb{R}^{q \times d}, \mathbf{w}_C \in \mathbb{R}^{c \times d}, \mathbf{w}_P \in \mathbb{R}^{p \times d}$.

3.2 Matching Layer

After obtaining the word representation, word-level attention is added to enhance the word representation. Given two word embedding sequences $\mathbf{w}_U, \mathbf{w}_V$, word level attention is calculated as:

$$\begin{aligned} \mathbf{M}'_{UV} &= \mathbf{w}_U \cdot \mathbf{w}_V^\top \\ \mathbf{M}_{UV} &= \mathrm{softmax}(\mathbf{M}'_{UV}) \\ \mathbf{w}_U^V &= \mathbf{M}_{UV} \cdot \mathbf{w}_V, \end{aligned} \tag{1}$$

where $\mathbf{M}'_{UV}$ contains dot products between each word in $\mathbf{w}_U, \mathbf{w}_V$; softmax is applied on $\mathbf{M}'_{UV}$ row-wise.

Here three kinds of attention are calculated using eq. (1): 1) question-aware passage representation $\mathbf{w}_P^Q \in \mathbb{R}^{p \times d_w}$ 2) question-aware choice representation $\mathbf{w}_C^Q \in \mathbb{R}^{c \times d_w}$ 3) passage-aware choice representation $\mathbf{w}_C^P \in \mathbb{R}^{c \times d_w}$.

3.3 Sequence Modeling Layer

To model the contextual dependency of each text sequence, we use BiLSTMs to process the word representations obtained from the input layer and matching layer:

$$\begin{aligned} \mathbf{h}^q &= \mathbf{BiLSTM}\mathbf{w}_Q \\ \mathbf{h}^c &= \mathbf{BiLSTM}[\mathbf{w}_C; \mathbf{w}_C^P; \mathbf{w}_C^Q] \\ \mathbf{h}^p &= \mathbf{BiLSTM}[\mathbf{w}_P; \mathbf{w}_P^Q], \end{aligned} \tag{2}$$

where $\mathbf{h}^q \in \mathbb{R}^{q \times l}, \mathbf{h}^c \in \mathbb{R}^{c \times l}, \mathbf{h}^p \in \mathbb{R}^{p \times l}$ are the hidden states of the BiLSTMs.

3.4 Attention Layer

To further focus on the important words in the text, self attention is applied on $\mathbf{h}^q, \mathbf{h}^c$ to obtain the final question and choice representations $\mathbf{q} \in \mathbb{R}^l$ and $\mathbf{c} \in \mathbb{R}^l$. Specifically, the self attention for $\mathbf{h}^q$ is calculated as:

$$\begin{aligned} \alpha &= \mathrm{softmax}(\mathbf{h}^q \cdot \mathbf{W}^\top) \\ \mathbf{q} &= \mathbf{h}^{q\top} \alpha, \end{aligned} \tag{3}$$

where $\alpha \in \mathbb{R}^q$ is the weight of each word in question. Similarly, we obtain $\mathbf{c}$.

Finally, word-level attention is calculated between $\mathbf{h}^p$ and $\mathbf{h}^q$ (as in eq. (1)) to obtain a question-aware passage representation, which is used as the final passage representation $\mathbf{p} = \mathbf{h}^{pq}$.

3.5 Output Layer

For each tuple $\{\mathbf{q}, \mathbf{p}_k, \mathbf{c}_k\}$, two scores are calculated by matching 1) passage and choice 2) question and choice. Both are formulated using bilinear matching and a softmax function is applied over K choices to discriminate words between different choices:

$$\begin{aligned} s_k^{pc} &= \mathbf{p}_k \mathbf{W}^{pc} \mathbf{c}_k \\ s_k^{qc} &= \mathbf{q} \mathbf{W}^{qc} \mathbf{c}_k \\ \mathbf{s} &= \mathrm{softmax}(\mathbf{s}^{pc}) + \mathrm{softmax}(\mathbf{s}^{qc}), \end{aligned} \tag{4}$$

where s_k^{pc}, s_k^{qc} are the scores for choice k, $\mathbf{s}^{pc}, \mathbf{s}^{qc}$ are vectors of scores for all K choices and $\mathbf{s}$ contains the final scores for these choices. After obtaining the final prediction (i.e., probability over K choices), we use a cross-entropy loss to train the model.

3.6 Essential Term Selector

Given a question $\mathbf{Q}$ and K choices $\mathbf{C}_1, \ldots, \mathbf{C}_K$, the goal of the selector is to predict a binary variable y_i for each word Q_i in the question $\mathbf{Q}$, where y_i equals 1 if Q_i is an essential term (0 otherwise). To address the problem, ET-Net shares the same input layer, matching layer, and sequence modeling layer to obtain the hidden states $\mathbf{h}^q$ for question $\mathbf{Q}$ as shown in eq. (2).

The hidden states are then concatenated with the feature embeddings of $\mathbf{Q}$ again, and fed into a projection layer to obtain the binary vector for each word:

$$\mathbf{y} = [\mathbf{q}^q; \mathbf{f}^q] \cdot \mathbf{w}^s, \tag{5}$$

where $\mathbf{y} \in \mathbb{R}^q$ is the final prediction for each word in the question, $\mathbf{f}^q$ contains the feature embeddings of $\mathbf{Q}$, and $\mathbf{w}^s$ contains the learned parameters.

Given the prediction $\mathbf{y}$ and the ground-truth, we use a binary cross-entropy loss to train the model.

4 Experiments

In this section, we first evaluate the performance of ET-Net (i.e., the selector module) and then discuss the performance of our proposed model on the ARC dataset. The model is built on PyTorch and spaCy is used to preprocess the dataset.

4.1 Performance on Essential Term Selection

For the term selector, we use the same dataset as in Khashabi et al. (2017). We split the dataset into *Train*, *Dev* and *Test* sets using an 8/1/1 split.

Table 1: Experimental results of the selectors

Model	Precision	Recall	F1
MaxPMI	0.88	0.65	0.75
SumPMI	0.88	0.65	0.75
PropSurf	0.68	0.64	0.66
PropLem	0.76	0.64	0.69
ET Classifier	0.91	0.71	0.80
ET-Net	0.74	**0.90**	**0.81**

Table 2: Examples of essential terms predicted in the questions. Red colored is true positive while blue colored is false positive.

Example questions
Which unit of measurement can be used to describe the length of a desk ?
One way animals usually respond to a sudden drop in temperature is by
Organisms require energy to survive. Which of the following processes provides energy to the body ?

Table 1 shows the performance of our term selector and baseline models (result from Khashabi et al. (2017)). As we can see, ET-Net achieves a comparable result than the ET Classifier which needs a large amount of handcrafted features. ET-Net also shows promising performance in terms of recall, which is an important metric in query reformulation. Since missing any of the essential terms will make the question difficult to answer, it is important to ensure a high recall that will not discard any essential terms from the the reformulated query. Another merit of ET-Net is that it can be further combined with the neural network based reader as an end-to-end trainable model, which is not practical for the SVM based ET Classifier.

As shown in table 2, ET-Net is capable of selecting most of the ground-truth essential terms. Moreover, it neglects certain words (such as 'organisms') which have a high TF-IDF in the corpus but are not relevant to a particular question. This shows its ability to find out essential terms according to the context of the question.

4.2 Performance on ARC

We train two variants of ET-RR on the ARC dataset:

- ET-RR (Reader only). We concatenate the

Table 3: Experimental results on ARC dataset

Model	Dev	Test
Guess All (random)	/	25.02
BiDAF	/	26.54
DGEM	/	27.11
KG^2	/	31.70
BiLSTM Max-out	/	33.87
ET-RR (Reader only)	38.93	35.33
ET-RR	**43.29**	**36.61**

question and choice as a query for each choice, which is the same as other baselines.

- ET-RR. We use the essential term selector to pick out essential terms in the question, then concatenate the essential terms and the choice as a query for each choice.

The performance of our proposed model is shown in table 3. We directly report the result of other baselines from the leaderboard.[1] As shown, ET-RR with reader only is already able to outperform the state-of-the-art. This shows the efficiency of an attention-enhanced reader that can effectively focus on essential terms and discriminate between different choices. ET-RR (i.e., with the term-selector-aware retriever) further improves performance, leading to an 8.1% improvement over the state-of-the-art BiLSTM Max-out method.

Acknowledgments

We thank Liang Wang from Yuanfudao and Daniel Khashabi from University of Pennsylvania for helpful suggestions to re-implement their work.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue, Pavan Kapanipathi, Nicholas Mattei, Ryan Musa, Kartik Talamadupula, and Michael Witbrock. 2018. A systematic classification of knowledge, reasoning, and context within the arc dataset. In *QA@ACL*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *CoRR*, abs/1803.05457.

Peter A. Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T. Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *CoRR*, abs/1802.03052.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2017. Learning what is essential in questions. In *CoNLL*.

Heeyoung Kwon, Harsh Trivedi, Peter Jansen, Mihai Surdeanu, and Niranjan Balasubramanian. 2018. Controlling information aggregation for complex question answering. In *ECIR*.

Preslav Nakov, Lluís Màrquez i Villodre, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. Semeval-2016 task 3: Community question answering. In *SemEval@NAACL-HLT*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Liang Wang. 2018. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. In *SemEval@NAACL-HLT*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI*.

[1]Snapshot from `http://data.allenai.org/arc/` on August 27, 2018