

Learning to Read Academic Literature

Jialu Wang, Yining Hong, Runqing Zhou, Xinbing Wang

Shanghai JiaoTong University

800 Dongchuan RD. Minhang District, Shanghai, China

{faldict, evelinehong, zrql6sytu, xwang8}@sjtu.edu.cn

Abstract

Machine reading comprehension (MRC), which requires machines to answer questions about a given context, has attracted much attention in recent years. However, academic literature is still beyond the scope of state-of-the-art MRC systems, rendering an MRC task on academic literature strongly needed. In this paper, we propose PAPERQA, a novel dataset focusing on the corpus of research papers on machine learning. PAPERQA consists of over 12,000 question-answer pairs posed by crowdworkers on a set of over 1,800 academic abstracts. To better incorporate semantic information, we design a new model which utilizes the shared query aware context representation as the base of sentence ranking and answer extraction. Experimental evaluations show that our model outperforms state-of-the-art MRC models on this task. Our work helps to develop services on academic QA and benefits researchers by saving much time on paper scanning.

Introduction

With the rapid development of deep learning in recent years, the number of published papers shows an astonishing upward trend (15,400 machine learning research papers published in 2015, 32,300 published in 2016 and 54,600 published in 2017)¹, which requires researchers to spend a large amount of time reading innumerable academic papers for the purpose of research, e.g., for survey of related work. Although more than a few paper-reading groups have been formed to help highlight the most critical information in research papers, time-consuming human efforts are inevitably needed, motivating us to design an intelligent system as an alternative. Machine reading comprehension (MRC) systems (Hirschman et al. 1999), which enable machines to answer questions about a certain document with a thorough understanding of it, have become an accessible and productive substitute for human labor. Application of MRC to academic papers can save researchers much time on reading papers as well as sorting out papers that meet their requirements.

In this paper, we focus on utilizing the power of MRC to search as well as highlight the essential information in academic paper abstracts in certain topics. To achieve this goal, we start with training machines to perform question answering tasks, using the accuracy of answers to repre-

sent how machines read and comprehend. However, existing MRC datasets, such as SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018), CNN/Daily Mails (Hermann et al. 2015) and MS MARCO (Nguyen et al. 2016) mostly concentrate on news articles and stories. An alternative and distinctive dataset focusing on academic literature is strongly needed.

Inspired by this, we present a novel MRC dataset on academic abstracts called PAPERQA. PAPERQA consists of over 12,000 question-answer pairs based on a set of over 1,800 abstracts from machine learning papers. These papers have been accepted by top-tier machine learning conferences. The questions are proposed according to the specific context in each abstract, asking about the paper’s objective, method, model, experiment and others. Answers to these questions, consisting of spans (i.e., sequences of words) in the corresponding abstract, are annotated by students with machine learning background. The reasons why corpus in PAPERQA is constrained to research paper abstracts are twofold. First, since an abstract summarizes the main content of a paper, it precisely provides the most concerned information required by researchers, while neglecting trivial details that one normally shows minor interest in at the very first glimpse of the paper. Second, the long content of academic papers would be very different in structures and presentation of formulas, figures and tables, making it difficult to extract useful and concise information for further use.

Figure 1 shows a sample abstract in our dataset. For the question “*what problem does this paper study?*”, the answer is highlighted in the text and presented below as well. Typically, there are several question-answer pairs for a piece of abstract. From this figure, we can see that PAPERQA has three major characteristics that make it challenging and distinguishing. (1) It is a question answering dataset based on the corpus of academic abstracts. (2) Most of the questions require deep comprehension and reasoning beyond simple word matching or extraction. (3) The answers may contain sophisticated terminology. Such entities require external knowledge to recognize.

To tackle the challenging task and assess the difficulty of PAPERQA, we benchmark some existing machine comprehension models and propose an intuitive model based on sentence selection (Yu et al. 2014) and word labeling to evaluate their performances. Our model mainly consists of three modules: attention-based context and query representation, sentence ranking, and answer extraction. In this model, we

Multiple instance learning (MIL) is a variation of supervised learning where a single class label is assigned to a bag of instances. In this paper, we state the MIL problem as learning the Bernoulli distribution of the bag label where the bag label probability is fully parameterized by neural networks. Furthermore, we propose a **neural network-based permutation-invariant aggregation operator** that corresponds to the attention mechanism. Notably, an application of the proposed attention-based operator provides insight into the contribution of each instance to the bag label. We show empirically that our approach achieves comparable performance to the best MIL methods on benchmark MIL datasets and it outperforms other methods on **a MNIST-based MIL dataset and two real-life histopathology datasets** without sacrificing interpretability.

Question 1: What problem does this paper study?

Answer 1: **Multiple instance learning**

Question 2: What approach does this paper propose?

Answer 2: **a neural network-based permutation-invariant aggregation operator**

Question 3: What dataset does this paper use?

Answer 3: **a MNIST-based MIL dataset and two real-life histopathology datasets**

Figure 1: An example of question answering on abstracts.

apply bi-directional attention flow (Seo et al. 2016) on top of the concatenation of word embedding and character embedding, whose encoding vectors are shared by the other two modules. Next, we build a multi-layer perceptron to conduct a match score corresponding to each sentence in the abstract. The sentence of the highest score will be taken as the evidence. Finally, to extract the specific span of words, we pass the query-aware context representation into a biLSTM-CRF model and tag the answer sequence. Empirical results on our datasets of scientific paper abstracts show that our model significantly outperforms the baseline models.

In sum, our contributions could be summarized as follows. First, we provide a challenging machine comprehension dataset focusing on scholarly paper abstracts, quite different from other existing question answering datasets such as SQuAD (Rajpurkar et al. 2016). Second, we propose a novel model which outperforms other existing machine comprehension models in academic literature reading task. Finally, from the perspective of applications, our work helps to develop tools to efficiently identify the essential information in massive abstracts and could be used to establish an academic knowledge base where machine learning entities could be extracted from answers.

Dataset Construction

In this section, we introduce the process through which we establish the corpus, propose questions and collect answers via crowdsourcing.

Abstract Selection

The recent five years have witnessed an astounding growth in the field of machine learning, especially deep learning, which makes our dataset more meaningful to focus on machine learning papers. In addition, papers from different fields vary a lot in structures, contents, objectives, etc. Such distinctions make it hard for machines to learn multifarious patterns. Papers on machine learning share similar patterns, ensuring that the task is learnable. Under these considerations, we collect papers in the field of machine learning published after the year of 2012, mainly sourced from top-tier conferences, such as NIPS, ICML, ICLR, AAAI, etc. We extract each paper’s abstract along with its title, authors to build the base of our dataset. Abstracts that are too short are discarded.

Question Posing

We establish a question base composed of a finite number of questions. Questions are rendered through empirical observation of hundreds of abstracts, ensuring that they can be applied to numerous abstracts rather than only a few of them. For a given abstract, the proposed questions are selected from the question base and contingent on the specific context of the abstract. For instance, if a paper proposes a model, we ask about what the model is, what it is based on and how it outperforms previous models, etc; if an experiment is carried out, questions concern applied datasets and demonstrated results. The questions in the question base, divided into four types of semantic heading as Dernoncourt and Lee (2017)’s setting, *objective, method, results, and others*, are finite and general. This form of question posing is due to the fact that the more general questions we ask, the deeper comprehension of the abstract is needed. On the contrary, specific and non-unified questions focus on details related to the context, and answers can be retrieved using merely lexical or syntactic variation but not understanding of the academic knowledge. Thus, specific and non-unified questions do not distinguish our datasets from datasets like SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018) in that they do not ask about more abstruse academic knowledge, and do not require deeper understanding as well. An example of a question set that we provide according to a specific abstract is shown in Table 1, which is a subset of the question base.

Category	Question
Objective	What problem(s) does this paper address?
Method	What model does this paper propose?
Method	What is the proposed model based on?
Experiments	What does the result of this paper show?
Experiments	How does this result outperform existing work?

Table 1: An example of a question set for a specific abstract.

Answer Sourcing

We create an interactive crowdsourcing website, which randomly presents a paper abstract in our database with several questions following the abstract. We invite over 200 crowdworkers to assist in building this dataset. Our crowdworkers are college students majoring in computer science who have taken machine learning courses before. Students are

awarded bonus according to their performance. Each student provides answers in approximately 10 abstracts on average, and the maximum number of question-answer pairs provided by a single student is about 200. Crowdworkers answer questions after acquiring a thorough understanding of the abstract presented. They may render the answer null if the abstract contains insufficient information. That is, crowdworkers select questions that they can answer from the preparative question base, thus constructing a specific question set for each abstract. Answers can only be attained by highlighting and copying continuous words (i.e. span) from the abstract. We provide our crowdworkers with detailed instructions as well as examples of good and bad answers. The answers selected by crowdworkers can then be stored into our dataset.

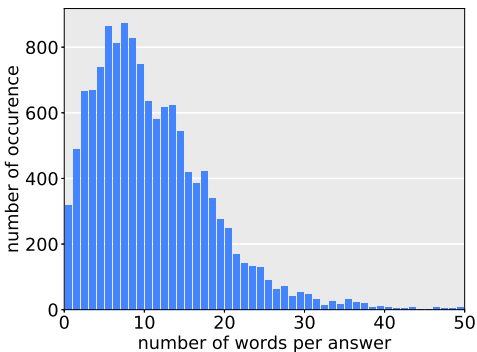


Figure 2: The distribution of the length of answers.

The final clean-up step is done through human efforts as well. To ensure the quality of this dataset as well as evaluate human performance, each answer is scrutinized by at least two crowdworkers. Crowdworkers examine whether the answer is valid, e.g., whether the answer makes sense, serves as an answer to the specific question and is of proper length. Valid answers are maintained. Answers either too short or too long are revised. Answers that are entirely nonsense are discarded and re-supplied by our crowdworkers.

Type	Objective	Method	Results	Others	Total
#(Number)	1783	4821	2564	3721	12889

Table 2: Number of QA pairs according to question types

Dataset Analysis

In total, we collect 1,892 abstracts and 12,889 question-answer pairs. Table 2 counts the number of QA pairs according to question types. *Method* covers nearly 40% of the total, 2.7 times as large as the percentage of *Objective*. Thus, types of questions are not that unbalanced in number. Figure 2 illustrates the distribution of answers’ lengths, where the majority are centralized between 2 and 20 approximately. Furthermore, we split our datasets into 3 parts as train/dev/test in an approximate ratio of 8 : 1 : 1. The train and dev sets are available online².

²<http://bit.ly/PaperQA>

Proposed Methodology

As the example in Figure 1 shows, our dataset would be challenging for models that focus on word matching but ignore intrinsic semantics such as (Wang and Jiang 2016a; Wang et al. 2016b; Tan et al. 2016). The result of QANet (Yu et al. 2018) listed in Table 3 supports this statement as well. QANet matches the word *model* but fails to clarify its semantic functions. We also assume that each sentence in an abstract typically serves as one semantic heading correlated to the questions, such as objective, method, and results. Thus, each sentence is relatively independent at the semantic level, which motivates us to match the sentences with questions by attention mechanism (Vaswani et al. 2017). Following this idea, we intend to first find an evidence sentence where the answer sequence most likely appears in the semantic level, and then extract a span from the evidence snippet as the final answer. Based on this analysis, we design a two-stage framework consisting of sentence ranking (Wang and Nyberg 2015) and answer extraction, as is illustrated in Figure 3, both of which share the context’s query aware vector representation acquired by attention mechanism. Our approach makes a deeper comprehension of academic abstracts without the utilization of external knowledge.

Problem Formulation

As the common setting of machine reading comprehension tasks, given an abstract with n sentences $P = \{s^{(1)}, s^{(2)}, \dots, s^{(n)}\}$, where each sentence $s = \{c_1, c_2, \dots, c_l\}$ has arbitrary length of l , and a question with m words $Q = \{w_1, w_2, \dots, w_m\}$, our task is to predict an answer A to question Q based on evidence provided by the abstract P . In the setting of our dataset, the answer A is constrained to a sequence of consecutive words $\{c_i, \dots, c_j\} \subseteq s^{(a)} (1 \leq i \leq j \leq l)$ as a span located in the a -th sentence of abstract P . Most questions involve semantic-level comprehension, and answers typically contain certain terminology. This challenges machine readers to have deeper comprehension, and also requires more reasoning on the abstract.

Question Aware Vector Representation

Similar to most existing question-answering models, we obtain the embedding x of each word w by concatenating its word embedding $x_w \in \mathbb{R}^{d_1}$ and character embedding $x_c \in \mathbb{R}^{d_2}$. The word embedding is initialized from the pre-trained GloVe (2014) word vectors of dimension $d_1 = 300$ and is fixed during training. To build a trainable character embedding, we use a bi-LSTM network to extract character-level features, taking the final hidden states and representing the word as $d_2 = 100$ dimensional character vectors. Finally, we obtain the word representation $x = [x_w; x_c] \in \mathbb{R}^{d_1+d_2}$.

Suppose each sentence s in the abstract and the question Q are converted to their word representations $H = \{x_t^p\}_{t=1}^l \in \mathbb{R}^{d \times l}$ and $U = \{x_i^q\}_{i=1}^m \in \mathbb{R}^{d \times m}$ separately where $d = d_1 + d_2$, we then apply bi-directional attention flow (Seo et al. 2016) to incorporate question information into sentence representation. Firstly, we compute the similarity matrix $S \in \mathbb{R}^{l \times m}$ by

$$S_{tj} = f(H_{:t}, U_{:j}) \in \mathbb{R} \quad (1)$$

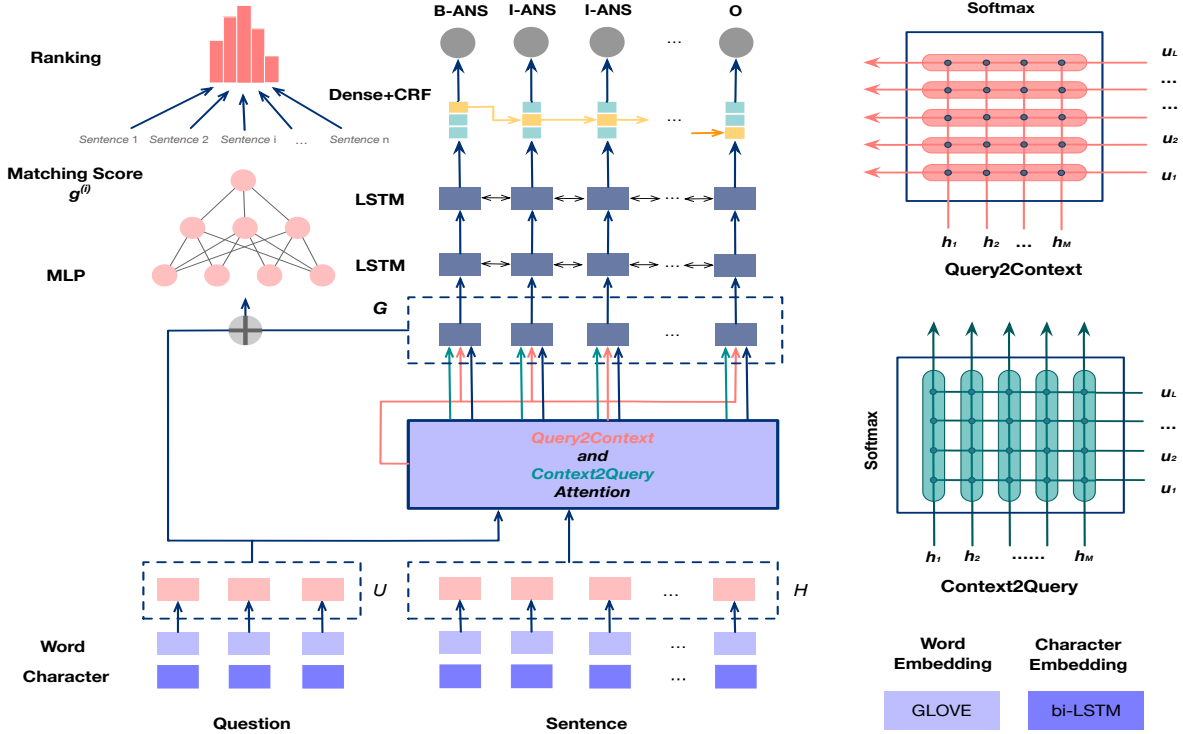


Figure 3: Overview of our model. At first, we concatenate word embedding and character embedding to form the word representation. Then biDAF is applied to get the context-to-query attention and query-to-context attention. Next, we use MLP to select the sentence with the highest rank as the evidence snippet. Finally, the query-aware context sequence is passed into a biLSTM-CRF model in order to extract the specific word span.

with the similarity function

$$f(h, u) = w_S^\top [h; u; h \circ u] \quad (2)$$

where w_S is a trainable weight vector and \circ is element-wise multiplication. Then the similarity matrix S is normalized by the softmax function across column as \hat{S} and across row as \tilde{S} respectively. We then compute the context-to-query attention by $\tilde{U} = U \cdot \hat{S}^\top \in \mathbb{R}^{d \times l}$ and the query-to-context attention by $\tilde{H} = H \cdot \tilde{S} \cdot \hat{S}^\top \in \mathbb{R}^{d \times l}$. Finally, a simple concatenation is used to yield the question-aware vector representation of each word in the sentence

$$G_{:t} = [H_{:t}; \tilde{U}_{:t}; H_{:t} \circ \tilde{U}_{:t}; H_{:t} \circ \tilde{H}_{:t}] \in \mathbb{R}^{4d} \quad (3)$$

where $G_{:t}$ refers to the t -th column vector (corresponding to t -th context word) of $4d$ dimension. The conducted query aware context representation G would be shared by both sentence ranking and answer extraction later.

Sentence Ranking

We first locate the sentence where the answer most likely appears, which is also called question answering sentence ranking (QASR). Instead of using the word representation directly, we exploit the query aware context representation for a better understanding of the sentence semantics. To be specific, the question aware representation of each sentence $G^{(i)} = \{G_{:t}^{(i)}\}_{t=1}^l$ and the question representation $U = \{x_t^q\}_{t=1}^m$ are added up separately, combined and passed

into a multi-layer perceptron (MLP) with one hidden layer, denoted as β , for a match score $g^{(i)}$.

$$g^{(i)} = \beta \left(\sum_{t=1}^l G_{:t}^{(i)}, \sum_{t=1}^m x_t^q \right) \quad (4)$$

We use the following normalization function to represent the probability $g^{(i)}$ that the sentence $s^{(i)}$ contains the answer to the question Q :

$$\hat{g}^{(i)} = \frac{\exp(g^{(i)})}{\sum_{i=1}^n \exp(g^{(i)})} \quad (5)$$

We then rank the normalized match scores and take the sentence with highest score as the evidence. Hence, the corresponding loss function is

$$L_{SR} = - \sum_{i=1}^n (y_i \log \hat{g}^{(i)} + (1 - y_i) \log(1 - \hat{g}^{(i)})) \quad (6)$$

where $y_i \in \{0, 1\}$ denotes the label indicating whether the answer exists in the sentence. There is always only one $y_j = 1$ among all the sentences for a pair of abstract P and question Q .

Answer Extraction

Once the evidence sentence is extracted, the next step is to pinpoint specific word sequence as the final answer. We model this as a sequence tagging problem (Yao et al. 2013)

rather than only predicting the start and end points of answers (Wang and Jiang 2016b), because the sequence model could take more syntactic structure information into consideration. In sequence tagging, each token in the sentence is tagged as one of the following labels: B-ANS (the beginning of answer), I-ANS (inside of answer), O (outside the answer). We use two layers of bi-directional LSTM (Hochreiter and Schmidhuber 1997) over the candidate sentence’s question aware representation to capture the semantic information, and also a fully connected layer to decode the score vector s_t

$$\begin{aligned} h_t &= \text{bi-LSTM}(h_{t-1}, G_{:t}) \\ h'_t &= \text{bi-LSTM}(h'_{t-1}, h_t) \\ a_t &= Wh'_t + b \end{aligned} \quad (7)$$

We adopt Huang, Xu, and Yu (2015)’s strategy to make use of neighbor tag information, and use a linear-chain CRF (Lafferty, McCallum, and Pereira 2001) model to conduct CRF scores $C(y_1, y_2, \dots, y_l)$ associated with the tag sequence $\{y_1, y_2, \dots, y_l\}$. Denoting y as the abbreviated symbol of tag sequence y_1, y_2, \dots, y_l , we apply a softmax to the scores of all possible sequences, in which the answer tokens are consecutive because we must ensure every answer is a span, to compute its possibility

$$P(y) = \frac{\exp C(y_1, y_2, \dots, y_l)}{\sum_{y_1, y_2, \dots, y_l} \exp C(y_1, y_2, \dots, y_l)} \quad (8)$$

And the loss function is defined as

$$L_{AE} = -\log P(\hat{y}) \quad (9)$$

where \hat{y} is the labeled tag sequence.

Joint Learning

For a given training sample, each sentence $s^{(i)}$ is fed into our model along with the question Q for a match score $\hat{g}^{(i)}$ and a predicted sequence $y^{(i)}$. The tags for sentences that don’t contain any answer are all labeled to O. Similar to Sultan, Castelli, and Florian (2016)’s work, the whole model is trained by minimizing the joint objective function

$$L = \gamma L_{SR} + \sum_{i=1}^n L_{AE}^{(i)} \quad (10)$$

where γ is a hyper-parameter for tuning the weight of two loss functions.

Experiment

In this section, we benchmark our method on PAPERQA dataset and compare its performance with that of two machine reading comprehension models.

Implementation Details

We use NLTK³ tokenizer to preprocess the data. The maximum sentence length is set to 80 while the maximum question length is set to 16, ensuring that all data don’t exceed

³<https://www.nltk.org/>

these lengths. We batch training samples in the same abstract by length and dynamically pad the short sentences with a special symbol <PAD>. For word embedding, we use the pretrained 300 dimensional GLoVe (Pennington, Socher, and Manning 2014) word vectors which are fixed during training, while all the out-of-vocabulary words are replaced with a special symbol <UNK>, whose embedding is updated during training. Each character embedding is randomly initialized as a 100 dimensional trainable vector. For sentence ranking, the number of units of hidden layer is 128 and tanh is utilized as the activation function in the MLP. The hidden vector size in bi-LSTM for answer extraction is set to 300. We also apply dropout between layers with a dropout rate (2014) of 0.5. The hyper-parameter γ is 1.5, making the sentence ranking loss cover a larger weight relative to answer extraction loss. To train this model, we use the Adam (2014) optimizer with a learning rate of 0.001, in which exponential moving average is applied on all trainable variables with a decay rate 0.9.

Baselines

We conduct experiments with following models as baseline models.

R-NET R-NET (Wang et al. 2017) is an end-to-end neural network model for question answering task with the formulation of MRC. R-NET first matches the question and the passage with gated attention-based recurrent networks to obtain question-aware passage representation. Then a self-matching attention mechanism is employed to refine the representation by matching the passage against itself. Finally, the pointer networks are applied to locate the positions of answers from the passages. We use the NLPLearn implementation⁴.

QANet QANet (Yu et al. 2018) is the state-of-the-art Q&A architecture which takes first place in SQuAD leaderboard⁵. The encoder of QANet consists exclusively of convolution and self-attention, where convolution models local interactions and self-attention models global interactions. We use NLPLearn implementation⁶ with 740k trainable parameters.

Human Performance We evaluate human performance on PAPERQA’s dev and test sets. Recall that answers are scrutinized by at least two crowdworkers during the clean-up stage. We regard answers provided by crowdworkers in the clean-up stage as ground-truth answers, and treat original answers as human predictions.

Results

To evaluate the performance of different models on proposed dataset, we use two metrics. The Exact Match (EM) metric measures the percentage of predictions that match the ground truth answers exactly. The F1 score metric is a less strict metric measuring the overlap between the prediction and the ground truth answers. Both two metrics ignore punctuations and articles (such as *a*, *an* and *the*).

⁴<https://github.com/NLPLearn/R-net>

⁵<https://rajpurkar.github.io/SQuAD-explorer/>

⁶<https://github.com/NLPLearn/QANet>

Abstract	We investigate the potential of a restricted Boltzmann Machine (RBM) for discriminative representation learning. By imposing the class information preservation constraints on the hidden layer of the RBM, we propose a Signed Laplacian Restricted Boltzmann Machine (SLRBM) for supervised discriminative representation learning. The model utilizes the label information and preserves the global data locality of data points simultaneously. Experimental results on the benchmark data set show the effectiveness of our method.
Question	What model do the authors propose?
R-NET	a Signed Laplacian Restricted Boltzmann Machine (SLRBM) for supervised discriminative representation learning
QANet	The model utilizes the label information and preserves the global data locality of data points simultaneously.
PaperQA	a Signed Laplacian Restricted Boltzmann Machine (SLRBM)

Table 3: The comparison among RNET, QANet and our framework on a sample abstract.

Model	Dev		Test	
	EM	F1	EM	F1
R-NET (Wang et al. 2017)	11.57	32.51	11.28	31.69
QANet (Yu et al. 2018)	17.93	50.31	18.19	51.26
Our Model	19.80	55.03	19.23	54.55
Human Performance	62.41	87.56	63.76	86.98

Table 4: Experiment Results

Table 4 illustrates the performance of our model and the aforementioned baseline models on both dev and test set, as well as human performance. Our model achieves EM scores of 19.80/19.23 and F1 scores of 55.03/54.55 on dev/test set respectively, which beats all the other baseline models. However, there is still a significant gap between our model and human performance. We do notice that EM scores are generally much lower than F1 scores. This is because entities in academic abstracts are often modified by several complex words and clauses, which increases the difficulty of discriminating boundary words and presents a huge challenge of our dataset.

One point of interest is to examine how the performance of our model varies across the lengths of predicted answers. As is shown in Figure 4, our model performs stably and well in a wide range of answer lengths. We also note that there is performance degradation when the answer is either too short or too long. This is partly due to the intuitive nature of the evaluation metric.

Moreover, we observe how the performance of each model varies with respect to question types. In Figure 5, the height of each bar represents the F1 score. Our model, outperforming the other baseline models in each type except *Results*, is adept at *Objective* and *Method* questions but struggles with *Results* and *Others* questions. QANet is skilled at *Objective* and *Results* but surprisingly takes a poor performance on *Method*. The performance of R-NET is fairly balanced.

Furthermore, in order to get a further understanding of three models’ robustness and generalization, we collect some latest paper abstracts from Arxiv⁷ and run the pre-trained models. Our hands-on evaluation indicates that, typically, all three models can catch the core entities. As the sample result shown in Table 3, all three models seem to predict the answers with semantic correlation. However, the answer of QANet is only supplementary information intro-

ducing the model’s feature and advantage, not the model itself. This indicates that QANet might be fooled by the word *model*. In fact, same as QANet, sequence tagging module of our model also marks the whole sentence as a candidate answer. However, the sentence ranking module gives the previous sentence a higher matching score, leading to dispose of this candidate answer. Both our model and R-NET catch *SLRBM*, the name of the proposed model, while R-NET supplies more details.

Error Analysis

We analyze 50 error examples generated by our model from the test set. We identify four key factors in causing the errors, which are elaborated as below.

Incorrect Sentence In 16% of the examples, the predicted best-matched sentence is incorrect due to semantic-level similarity with the target sentence. For example, as is shown in Table 3, QANet generates a wrong sentence, and our model also makes similar mistakes in some cases.

Syntactic Complications and Ambiguities In 38% of the examples, our model generates sequences containing the same entity as the one that may exist in the correct answer. For instance, for the question “*What method does this paper propose?*”, some spans contain words like *approaches* and *methods* etc, which do not exactly refer to the real methods proposed in the papers, but are sometimes marked as answers as well.

External Knowledge In 4% of the examples, external knowledge is indispensable for answering the questions. Some complex terminologies are still too hard for machines to comprehend. A potential solution is to leverage knowledge base in question answering (Wang et al. 2016a).

Imprecise Boundaries The rest of errors consist of imprecise boundaries where one or more words are either missed or appended at the edge of the correct span. The majority of cases contain an extraneous verb such as *propose* and *present*. In other cases, the entire phrase/clause after a conjunction (e.g., *and / or*) is lost. For instance, for the question “*What method does this paper propose?*”, the correct answer is “*a method for object detection and recognition*”. However, our model generates a less favored answer, which is “*a method for object detection*”.

⁷<https://arxiv.org>

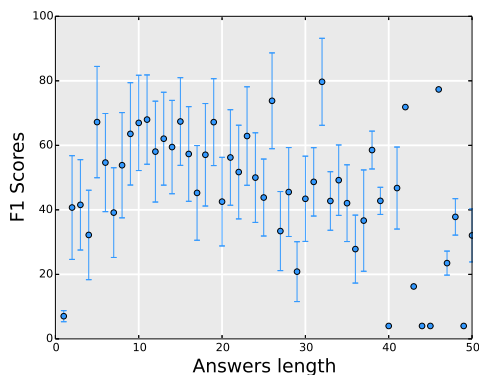


Figure 4: Performance of our model across lengths of answers. The blue dot indicates the mean F1 score at given length. The vertical bar represents the standard derivation of F1s at a given length.

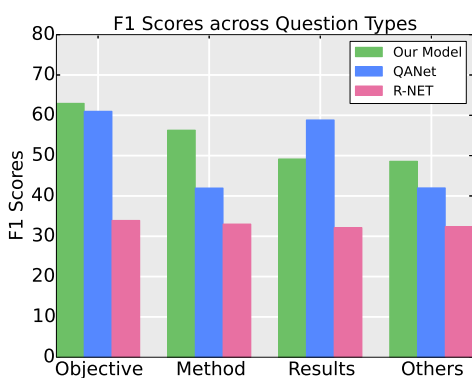


Figure 5: Performance of three models across question types. The height of each bar represents the F1 score.

Related Work

In this section, we introduce the prior work from the perspective of both datasets and machine reading comprehension models.

Benchmarking Datasets Reading Comprehension datasets require systems to identify a span in a text to answer a given question, which typically involves extracting relevant entities and reasoning based on rules. There have been several reading comprehension datasets up to present. MCTest (Richardson, Burges, and Renshaw 2013) contains 660 stories. Most of the stories and sentences are short, and the size of vocabulary is quite small as well. SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018), the most famous challenge in the field of question answering, contains about 100K question-answer pairs from 536 articles, where the context for each question is a single paragraph in these articles. CNN and Daily Mail QA datasets (Hermann et al. 2015) are two large-scale cloze datasets which contain numerous documents. MS Marco (Nguyen et al. 2016) is another MRC dataset sampled from real web documents and user queries. Close scrutiny of existing reading comprehension datasets, however, reveals that these datasets do not get involved in the corpus of academic papers. Such corpus presents a more challenging task demanding higher-level

intelligence for machines.

At present, there have been several works focusing on the domain of academic literature. The PubMed 200k RCT (Dernoncourt and Lee 2017) is a public large-scale dataset for sequential sentence classification built upon academic abstracts. However, this task is simple without requirements for machine comprehension and reasoning. (Cohan et al. 2018) summarizes scientific papers to abstracts and provide two datasets derived from Arxiv and PubMed. However, abstractive summarization is more like information retrieval and lack of comprehension. DLPaper2Code (Sethi et al. 2018) extracts and understands deep learning design flow diagrams and tables in a research paper and converts them into execution ready source code. SCITAIL (Khot, Sabharwal, and Clark 2018), also focusing on scientific QA task, treats multiple-choice question-answering as an entailment problem. It is constructed solely from natural sentences, which is quite different from our reading comprehension dataset. To the best of our knowledge, PAPERQA is the first dataset bringing machine comprehension into the corpus of academic abstracts.

Reading Comprehension Models A great number of end-to-end neural network models have been investigated to tackle the task of machine reading comprehension, including R-Net (Wang et al. 2017), DCN (Xiong, Zhong, and Socher 2016), ReasoNet (Shen et al. 2018), GA Reader (Dhingra et al. 2017) and QANet (Yu et al. 2018). They typically consist of an embedding layer, an encoding layer to integrate contextual information, an attention layer to incorporate query and context, a decode layer and an output layer which varies according to the specific QA task. However, in terms of our dataset, empirical observations indicate that word and pattern similarity often misleads the model, contributing to the answer located in a sentence with semantic correlation, which is not supposed to be the location for the right answer. Question understanding and adaption (Zhang et al. 2017) explores different question encoding, but it doesn't adapt to our dataset due to the questions' simple pattern in our dataset. DCR (Yu et al. 2016) extracts and ranks a set of answer candidates, while we take advantage of semantic information in the sentence level. S-Net (Tan et al. 2018) and (Wang and Nyberg 2015) takes advantage of joint learning, inspiring us to design a similar framework (Wang and Nyberg 2015) for end-to-end training.

Conclusion and Future Work

This paper is a first attempt at teaching machine to read and comprehend scholarly paper abstracts. We provide a new machine reading comprehension dataset alongside a challenging task. Then we propose a novel model to solve this task, composed of sentence ranking and sequence tagging stages, and end-to-end trained by joint learning. Empirical evaluations show that our model outperforms the state-of-the-art question answering models. We hope our work will benefit researchers in automatic paper survey, and the release of our dataset encourages further exploration. Simultaneously we will expand the size of our dataset with quality guaranteed.

References

- Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*.
- Dernoncourt, F., and Lee, J. Y. 2017. Pubmed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, 308–313.
- Dhingra, B.; Liu, H.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. In *ACL*.
- Hermann, K. M.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. *CoRR* abs/1506.03340.
- Hirschman, L.; Light, M.; Breck, E.; and Burger, J. D. 1999. Deep read: A reading comprehension system. In *ACL*.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9:1735–1780.
- Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *CoRR* abs/1508.01991.
- Khot, T.; Sabharwal, A.; and Clark, P. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- Lafferty, J. D.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.
- Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. Ms marco: A human generated machine reading comprehension dataset. *CoRR* abs/1611.09268.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.
- Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR* abs/1606.05250.
- Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.
- Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, 193–203. *ACL*.
- Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.
- Sethi, A.; Sankaran, A.; Panwar, N.; Khare, S.; and Mani, S. 2018. Dlpaper2code: Auto-generation of code from deep learning research papers. *CoRR* abs/1711.03543.
- Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; and Zhang, C. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. *CoRR* abs/1709.04696.
- Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.
- Sultan, M. A.; Castelli, V.; and Florian, R. 2016. A joint model for answer sentence ranking and answer extraction. *TACL* 4:113–125.
- Tan, M.; dos Santos, C. N.; Xiang, B.; and Zhou, B. 2016. Improved representation learning for question answer matching. In *ACL*.
- Tan, C.; Wei, F.; Yang, N.; Du, B.; Lv, W.; and Zhou, M. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. In *AAAI*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. 5998–6008.
- Wang, S., and Jiang, J. 2016a. Learning natural language inference with lstm. In *HLT-NAACL*.
- Wang, S., and Jiang, J. 2016b. Machine comprehension using match-lstm and answer pointer. *CoRR* abs/1608.07905.
- Wang, D., and Nyberg, E. 2015. A long short-term memory model for answer sentence selection in question answering. In *ACL*.
- Wang, B.; Guo, S.; Liu, K.; He, S.; and Zhao, J. 2016a. Employing external rich knowledge for machine comprehension. In *IJCAI*.
- Wang, Z.; Mi, H.; Hamza, W.; and Florian, R. 2016b. Multi-perspective context matching for machine comprehension. *CoRR* abs/1612.04211.
- Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *ACL*.
- Xiong, C.; Zhong, V.; and Socher, R. 2016. Dynamic coattention networks for question answering. *CoRR* abs/1611.01604.
- Yao, X.; Durme, B. V.; Callison-Burch, C.; and Clark, P. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*.
- Yu, L.; Hermann, K. M.; Blunsom, P.; and Pulman, S. G. 2014. Deep learning for answer sentence selection. *CoRR* abs/1412.1632.
- Yu, Y.; Zhang, W.; Hasan, K. N.; Yu, M.; Xiang, B.; and Zhou, B. 2016. End-to-end answer chunk extraction and ranking for reading comprehension.
- Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR* abs/1804.09541.
- Zhang, J.; Zhu, X.-D.; Chen, Q.; Dai, L.-R.; Wei, S.; and Jiang, H. 2017. Exploring question understanding and adaptation in neural-network-based question answering. *CoRR* abs/1703.04617.