# Learning to Read Academic Papers

**Yining Hong, Jialu Wang**
Shanghai JiaoTong University
{evelinehong, faldict}@sjtu.edu.cn

## Abstract

We present *PaperQA*, a challenging dataset of over 6000 human-generated question-answer pairs concerning academic knowledge. Crowdworkers supply questions and answers based on a set of over 1,000 abstracts from deep learning papers, with answers consisting of spans of text from the corresponding abstracts. This dataset is aimed at helping machines learn to read academic papers. We collect this dataset through a four-stage process designed to solicit exploratory questions that require reasoning. Then we propose a semantic segmentation model to solve this task and evaluate it on our dataset. We also propose an unsupervised named entity recogtion model to extract the specific entities in the abstracts. Finally, we build a website to show results of our model.

## Introduction

Teaching machine to read is a non-negligible part of 'True AI', people are making progress since the renaissance of deep learning, however, were not even close, the state-of-the-art models still hard to beat a human kid. Teaching machine to read paper is an even more untouchable dream. To challenge this task, we can start with training machines to do reading comprehension questions, like a child, and use the accuracies of question answers to indirectly represent how machines read and comprehend, which is smart because we need some metrics to evaluate.

Nowadays, there are several medias in China which provide latest news about machine learning papers, such as PaperWeekly and so on. In order to extract the most important information from these papers, paper reading groups are formed. However, this requires a large amount of human resource. We intend to replace human resource with machine in this process, and use machine to present some important information for us based on machine reading comprehension. To do so, we first need a machine reading comprehension dataset based on papers.

In this paper, we present a novel dataset for machine reading comprehension on academic abstracts: *PaperQA*. *PaperQA* consists of over 6,000 question-answer pairs based on a set of over 1,000 abstracts from machine learning papers, including papers accepted by top machine learning conferences(such as AAAI, NIPS, CVPR, ICCV, ICML, ACL, ECCV and EMNLP), and papers submitted on arXiv.org. The questions are fixed in each abstract, concerning objectives, methods, models, experiments and others. Answers to these questions consist of spans of the corresponding abstract that are highlighted by students of machine learning background.

The purpose of releasing *PaperQA* is twofold. First, by releasing this dataset, we propose a novel machine reading comprehension task on papers. Second, from an application perspective, it provides researchers with a tool to efficiently identify the most important information in a paper and decide whether to continue with the paper or not.

Some characteristics of *PaperQA* that make it challenging and distinguish it from prior machine reading datasets are listed as follows:

- It is a machine reading comprehension based on academic papers, which requires machines to learn prior knowledge.
- Some of the questions require reasoning beyond simple sentence-level or word-level analysis.
- The answer to each question is a span (i.e., sequence of words) of arbitrary length.
- Some questions have no answer in the corresponding article (the null span).

In this paper, we describe the dataset collection process. To assess the difficulty of *PaperQA*, We propose a baseline model based on sentence-level classification and word-level classification to, and evaluate its performance on our dataset. We also propose an unsupervised named entity recogtion model to extract the specific entities in the abstracts. To set an example of how our dataset can assist in academic research, we build a website called *AceNews* which extracts important information of newest machine reading papers, and presents comprehension of these papers. Moreover, we recommend papers for different users based on the information extracted and user behaviours. All of the above is accomplished by machine.

| Dataset | Sources | Formulation |
|---|---|---|
| **PaperQA** | machine learning abstracts | spans in abstract |
| MCTest (Richardson, Burges, and Renshaw 2013) | stories | MRC multiple choice |
| CBT (Hill et al. 2015) | stories from children's book | MRC cloze |
| CNN/Daily Mail (Hermann et al. 2015) | CNN news | MRC cloze |
| SQuAD (Rajpurkar et al. 2016) | Wikipedia articles | MRC spans in passage |
| PubMed (**?**) | medical abstracts | sentence classification |

Table 1: A survey of several reading comprehension datasets and datasets of abstracts. *PaperQA* is the only MRC dataset consisting of academic abstracts.

## Related Work

We start with a survey of existing machine reading comprehension datasets and datasets of abstracts, which vary in sources, size, difficulty, collection methodology and format of answers. We discuss about various sources of articles and task formulation in these datasets (see Table1 for an overview).

### Machine Reading Comprehension Datasets

**Machine Comprehension Test (MCTest)** This is a dataset from MSR, which contains 660 stories, each story has 4 human asked questions (Natural Language Question), and for each question, therere 4 candidate answers. This is pretty much like reading comprehension questions for pupils. Most of the stories are short and sentences are fairly short as well, and the size of vocabulary is small.

**Childrens Book Test (CBT)** A dataset from FAIR, which contains stories from childrens books. Each story in this dataset is a 20 consecutive sentences from childrens books, and remove a word from the consecutive 21st sentence, as the question, or query. Therere 4 splits of this dataset which are classified by the distinct types of word removed in queries: Named Entities, Common Nouns, Verbs, Prepositions. This type of fill in the blank query is called Cloze type question. For each question, therere 10 candidate answers which taken from the story, and all have same POS with the correct answer word.

**CNN/Daily Mail** *CNN/Daily Mail* QA dataset is released by Google DeepMind, which the largest (AFAIK) QA dataset. *CNN* dataset contains over 90K of of CNN news, and averagely has 4 queries per story, which gives 380K of story-question pairs; *Daily Mail* has about 200K new stories, and also, each story has 4 queries, which totally gives 880K story-question pairs.

**The Stanford Question Answering Dataset (SQuAD)** This dataset is recently released by Stanford University, which contains about 100K of question-answer pairs from 536 articles, the story for each question is a paragraph from these articles. Questions in *SQuAD* dataset are generated by crowdworkers so theyre NLQ. The formulation of answers is spans of the passage, which is similar to *PaperQA*.

### Datasets of Abstracts

**PubMed 200k RCT** It is a dataset based on for sequential sentence classification. The dataset consists of approximately 200,000 abstracts of randomized controlled trials, totaling 2.3 million sentences. Each sentence of each abstract is labeled with their role in the abstract using one of the following classes: background, objective, method, result, or conclusion. *PubMed 200k* is similar to *PaperQA* in that they are both datasets of abstracts, and the categories of questions *PaperQA* are much like the classes in *PubMed 200k*. However, *PaperQA* searches for more specific answers, which are spans rather than sentences, thus increasing the difficulty. Moreover, *PaperQA* is based on machine learning papers while *PubMed 200k* is based on medical papers.

## Dataset Construction

We collected *PaperQA* in four stages: paper curation, question posing, answer sourcing, and dataset cleanup. These steps are detailed as follows.

### Paper Curation

In order to make sure our task is learnable, our data must follow a specific pattern. However, papers of different fields and time periods vary in objectives, contents and structures. Distinctions in abstract structures make it hard for machine to learn such different patterns. It is also impossible for us to raise general questions which go for diverse abstracts. We look for papers following similar patterns. The recent five years have seen an increase in papers on machine learning, especially on deep learning. These papers have similar abstract structures. Given this advantage, we use papers on machine learning published after the year of 2012. To retrieve high-quality paper abstracts, we use most cited papers in top AI conferences (AAAI, ICCV, ECCV, EMNLP, NIPS, CVPR, and ACL).

## Question Posing

We intend to extract the most important information in papers. The information we want is similar in each abstract, mainly about objective, what is proposed, experiment and its result. Therefore, The questions are fixed for each paper abstract. The questions are divided into three categories: objective, method, and experiment. In the objective part, we ask about the aim of the paper and the problem addressed. In the method part, we ask about what is proposed. As several things can be proposed in a paper, including method, model, algorithm, framework and dataset etc. We set a checkbox allowing crowdworkers to select the specific item proposed. For each item proposed, we ask about what it is based on how it differs from previous method/model/algorithm etc. Finally, in the experiment part, we ask about what experiment is carried out and the result.

## Answer Sourcing

We create an interactive crowdsourcing website, which randomly presents a paper abstract in our database with several questions. Crowdworkers answer questions according to the abstract. They may reject the question as nonsensical, or select the null answer if the abstract contains insufficient information. Answers can only be attained by highlighting and copying continuous words (or a span) from the abstract. We provide our crowdworkers with detailed instructions as well as examples of good and bad answers. The answers can then be stored in our database. Our crowdworkers are students in Shanghai Jiaotong University who have taken machine learning classes. The students are required to answer 8 questions in each abstract.

## Dataset Cleanup

After collecting question-answer pairs, we filter the answers too short and check them manually. To our surprise, the filled answers show strong professional skills with high quality. To obtain a dataset of the highest possible quality we use a validation process that mitigates issues. We examine the dataset by ourselves to leave out some obviously wrong answers. Then we put every abstracts along with all the non-empty question-answer pairs in a json file. Finally our dataset contains 1,030 abstracts and 8,374 question-answer pairs.

## Dataset Structure and Dataset Analysis

### Dataset Structure

**Dataset Analysis** Table 2 counts the number of answers per question and shows their category. Several experiment results are allowed for one paper, so there are 1171 answers for the question "What experiment does this paper carry out to evaluate the result?", exceeds the total number of abstracts. This table indicates that our dataset is not excessively unbalanced.

# Methods and Experiments

We use a three-step approach to get answers. The first is a sentence classification model which categorizes sentences into four answer types. The second is a POS-attached sequence tagging model, which incorporates part-of-speech(POS) tags to select continuous words (or span) for our answer. The performance of these two supervised models are evaluated on our dataset. The third is an unsupervised named-entity-recognition model to extract the specific name of models, datasets or evaluation metrics in the abstracts.

## Sentence Classification

We summarize our question-answer pairs and divide them into three categories: *purpose*, *method* and *experiment*. Then, every abstract is divided into sentences. We check each sentence whether it contains any answer. If so, we label it with corresponding answer's category. If the sentence doesn't contain any answers, we label it as *others*. In total, we clean out 6,383 sentences with four-category labels. Some sentences have more than one labels because it contains answers to more than one questions. In this way, our first step is to take our task as a text classification problem.

We use the fastText (Joulin et al. 2017) model to deal with sentence classification, which uses a bag of n-grams as features and the hierarchical softmax as the linear classifiers. We set learning rate of as 0.1 to train models. We set size of word vectors as 100 and find that model performance is not sensitive to the size of word vectors. The training with 12 threads requires less than 100 epochs to converge and generally it takes less than 1 minute. Table 3 shows some sentences' predicted labels.

For evaluation metric, we plot the *precision-recall curves*. *Area under the curve(AUC)* is adopted as another quantitative measure. AUC in this paper refers to the area under the precision-recall curve.

Experimental Results:

## POS-Attached Sequence Tagging

Sentence classification locates the sentence where each answer lies. However, we must also select continuous words in this sentence as the final answer. In sentence classification, the label of each sentence indicates whether it contains an answer. In sequence tagging, each word is assigned a label, indicating whether it is part of an answer. If so, the word is labeled with the A and the number indicating to which question the answer is. For example, A4 represents that the word is part of an answer to the fourth question, and 0 represents the word is not part of an answer.

In such sequence tagging task, we have access to both past and future input features for a given time, we thus utilize a bidirectional LSTM network. By doing so, we efficiently make use of past features (via forward states) and future features (via backward states) for a specific time frame. We train bidirectional LSTM networks using backpropagation through time (BPTT). The forward and backward passes over the unfolded network over time are carried out in a similar way to regular network forward and backward passes, except that we need to unfold the hidden states for all time steps. We also need a special treatment at the beginning and the end of the data points. Thus, we use the CRF networks to make use of neighbor tag information in predicting current tags. Overall, we implement a biLSTM-CRF model (Huang, Xu, and Yu 2015).

| Category | Question | Numbers |
|---|---|---|
| Purpose | What is the objective/aim of this paper? | 963 |
| Purpose | What problem(s) does this paper address? | 857 |
| Methods | What method/approach does this paper propose? | 594 |
| Methods | What is this method based on? | 395 |
| Methods | How does the proposed method differ from previous methods/approaches? | 338 |
| Methods | What model does this paper propose? | 198 |
| Methods | What is this model based on? | 133 |
| Methods | How does the proposed model differ from previous models? | 122 |
| Methods | What algorithm does this paper propose? | 222 |
| Methods | What is this algorithm based on? | 143 |
| Methods | How does the proposed algorithm differ from previous algorithms? | 135 |
| Methods | What framework does this paper propose? | 120 |
| Methods | What is this framework based on? | 70 |
| Methods | How does the proposed framework differ from previous frameworks? | 61 |
| Methods | What dataset does this paper propose? | 61 |
| Experiments | What experiment does this paper carry out to evaluate the result? | 654 |
| Experiments | What does the result of this paper show? | 1171 |
| Experiments | How does this result outperform existing work? | 542 |

Table 2: Dataset Analysis

To further enhance the performance, we incorporate a pre-trained part-of-speech(POS) tagger and attach it to labels. Thus, the label of each word is composed of two parts: its POS tag, as well as an answer tag indicating whether it's part of an answer. Table 4 shows a sample sentence with each word tagged.

For evaluation metric, we use *Macro-averaged F1 score*. This metric measures the average overlap between the prediction and ground truth answer. We treat the prediction and ground truth as bags of tokens, and compute their F1. We take the maximum F1 over all of the ground truth answers for a given question, and then average over all of the questions. This POS-Attached biLSTM-CRF model reaches 86.7% F1 score on our dataset.

### Unsupervised Named Entity Recognition

## Conclusion

In this paper, we provide *PaperQA*, a QA dataset on academic paper abstracts, which contains more than 1,000 abstracts and 8,000 question-answer pairs. Then we propose a two-level framework to tackle this machine reading comprehension problem, and our model's performance is closed to human performance. We have made our dataset available freely to encourage more expressive models. Finally we build a website to interactively display the newest crawled arxiv paper abstracts and the important information answered by our model. The test result on the newest arxiv papers shows our model's generalization and robustness. Since the release of our dataset, we have already seen considerable interest in building models on this dataset, and the gap between our logistic regression model and human performance has more than halved. We expect that the remaining gap will be harder to close, but that such efforts will result in significant advances in reading comprehension.

## References


[1] 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.

[2] Hill, F.; Bordes, A.; Chopra, S.; and Weston, J. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR* abs/1511.02301.

[3] Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991.

[4] Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–431. Association for Computational Linguistics.

[5] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR* abs/1606.05250.

[6] Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text.

| Sentence | Prediction | Probability |
|---|---|---|
| While deep reinforcement learning has successfully solved many challenging control tasks, its real-world applicability has been limited by the inability to ensure the safety of learned policies. | others | 0.931695 |
| We propose an approach to verifiable reinforcement learning by training decision tree policies, which can represent complex policies (since they are nonparametric), yet can be efficiently verified using existing techniques (since they are highly structured). | purpose | 0.787732 |
| The challenge is that decision tree policies are difficult to train. | others | 1.000000 |
| We propose VIPER, an algorithm that combines ideas from model compression and imitation learning to learn decision tree policies guided by a DNN policy (called the oracle) and its Q-function, and show that it substantially outperforms two baselines. | methods | 0.973163 |
| We use VIPER to (i) learn a provably robust decision tree policy for a variant of Atari Pong with a symbolic state space, (ii) learn a decision tree policy for a toy game based on Pong that provably never loses, and (iii) learn a provably stable decision tree policy for cart-pole. | others | 0.568271 |
| In each case, the decision tree policy achieves performance equal to that of the original DNN policy. | experiments | 0.961477 |

Table 3: Sample sentences and their predicted results

| Word | Label |
|---|---|
| we | PR-0 |
| design | VB-0 |
| a | DT-A4 |
| time-weighted | JJ-A4 |
| reservoir | NN-A4 |
| sampling | VB-A4 |
| method | NN-A4 |
| to | TO-0 |
| maintain | VB-0 |
| and | CC-0 |
| update | VB-0 |
| ... | |

Table 4: Sample sentence with tagged words. The first part of a label is a POS tag.