Unsupervised Deep Learning based Multiple Choices Question Answering: Start Learning from Basic Knowledge

Chi-Liang Liu **Hung-yi Lee**
College of Electrical Engineering and Computer Science
National Taiwan University
{liangtaiwan1230, tlkagkb93901106}@gmail.com

Abstract

In this paper, we study the possibility of almost unsupervised Multiple Choices Question Answering (MCQA). Starting from very basic knowledge, MCQA model knows that some choices have higher probabilities of being correct than the others. The information, though very noisy, guides the training of an MCQA model. The proposed method is shown to outperform the baseline approaches on RACE and even comparable with some supervised learning approaches on MC500.

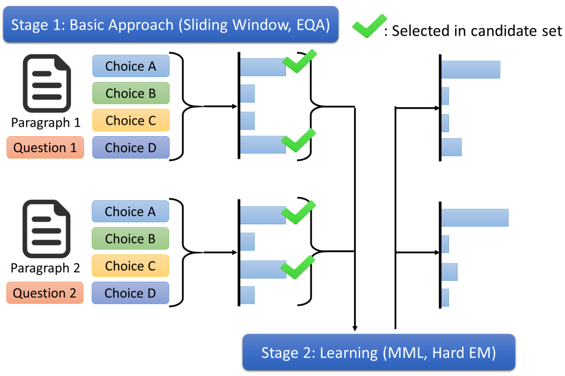

Figure 1: Overall training process.

1 Introduction

Question Answering (QA) has been widely used for testing Reading Comprehension. Recently, numerous question answering datasets (Weston et al., 2015; Rajpurkar et al., 2016, 2018; Yang et al., 2018; Trischler et al., 2017; Choi et al., 2018; Joshi et al., 2017; Kwiatkowski et al., 2019; Reddy et al., 2019; Richardson, 2013; Lai et al., 2017a; Khashabi et al., 2018) have been proposed. These datasets can be divided into two major categories: *Extractive Question Answering* (EQA) and *Multiple Choices Question Answering* (MCQA). In EQA, the answer has to be a span of the given reading passage, such as SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2017); while in MCQA, the answer is one of the given choices, such as MCTest (Richardson, 2013) and RACE (Lai et al., 2017a).

Recently, large pretrained language models such as BERT (Devlin et al., 2019) have exceeded human performance in some EQA benchmark corpora, for example, SQuAD (Rajpurkar et al., 2016). Compared to EQA, MCQA is not restricted the answer be spans in context. This allowed MCQA can have more challanging question than EQA including but not limited to logical reasoning or summarization. The performance gap between BERT and human performance is still significant. In this paper, we focus on MCQA.

A person who can read can deal with the MCQA task without too much further training, but this is not the case for a machine. The BERT-based models cannot be directly applied to solve the CQA task without seeing any MCQA examples. Even for the models achieving human level performance in EQA, they still need some MCQA examples with correct choices being labeled for finetuning.

The semi-supervised MCQA model training approach has been proposed (Chung et al., 2018), in which an initial MCQA model is used to answer the unlabelled questions to generate pseudo labeled data. Then pseudo labeled data is used to fine-tune the MCQA model to improve the performance. However, the initial MCQA model still needs some labeled examples to train. Also, Keskar et al. (2019); Raffel et al. (2020) proposed the unified model for EQA and MCQA model by unifying the multiple taks to span extraction task.

In this paper, we study the possibility of unsupervised MCQA Instead of starting from an initial MCQA model, here machine starts with some prior knowledge, including predefined rule or pretrained model. For example, a choice has higher probability to be correct, if there are more words overlapped between the document, the choice and the question. With the basic rule, the machine knows that some choices have higher probabilities of being correct than others, and some can be ruled out. With these basic rules, an MCQA model can be trained without any labeled MCQA examples.

With this approach, we got absolute gains of 4∼9% accuracy compared to the baseline methods on two MCQA benchmark corpora, RACE and MC500.

2 Unsupervised MCQA

We consider MCQA where we are given a question q, a passage p and a set of choices $C = \{c_1, c_2, ...c_n\}$, where n is the number of choices, and machine needs to select an answer $a \in C$.

We propose to address an unepervised MCQA in a two-stage approach (Figure 1). First, we pick the candidate set $\mathcal{T}$ from choices by fundamental rule from human knowledge (sliding window) or a model trained without MCQA data (EQA model). Second, we train a model to pick the final answer from the candidates.

2.1 Candidates Choosing

The candidate selection approaches give a score to each choice which represents the likelihood of being correct. We use two systems to calculate the scores, one using simple lexical features and another using a pre-trained EQA model. A choice is selected into candidate set $\mathcal{T}$ if the choice's score is higher than a threshold t, and is the top k scores among all the choices $\{c_1, c_2, ...c_n\}$ of a question q. In this way, each question has at most k candidates in $\mathcal{T}$. k should be smaller than n ($k < n$) to rule out the some less likely choice. A question will not have any choice in $\mathcal{T}$ if none of its choices pass the threshold t. Both t and k are the hyperparameters. Note that our methods do **not** gurantee the answer must in the candidate set and we do not need to choose candidates during inference.

Sliding Window (SW) We follow the sliding window algorithm in Richardson (2013), matching bag of words constructed from the question and choices to the passage to compute the scores of choices. The algorithm's details are shown in Algorithm 1.

EQA Matching In this setting, we use a pre-trained EQA model as our reference. Given a passage and a question, the EQA model outputs an answer A, which is a text span from the passage. Then we use a string-matching algorithm to compute the similarity between A and each candidate c, and the similarity serves as the score for each candidate. Gestalt Pattern Matching (Ratcliff and Metzener, July 1988) algorithm is the string-matching algorithm used here. The algorithm's details are shown in Algorithm 2.

Algorithm 1: Sliding Window

Input : Threshold t, max numbers of candidates k, a set of passage words P, set of words in question Q, and a set of words in choices $C_{1...n}$.
Define : $Count(w) \coloneqq \sum_i \mathbb{1}(P_i = w)$ where P_i is the i-th word in passage P;
Define : $IC(w) \coloneqq \log\left(1 + \frac{1}{Count(w)}\right)$
candidates ← Array[]
for $i = 1$ *to* n **do**
  $S \leftarrow C_i \cup Q$
  score$_i$ ←
$$\max_{j=1...|P|} \sum_{w=1...|S|} \begin{cases} IC(P_{j+w}), & \text{if } P_{j+w} \in S \\ 0, & \text{otherwise} \end{cases}$$
  if $score_i \geq t$ **then**
    candidates.append((i, score$_i$))
sort candidates descending by score
return first k elements of candidates

Algorithm 2: EQA Matching

Input : Threshold t, max numbers of candidates k, a set of passage words P, set of words in question Q, and a set of words in choices $C_{1...n}$ and a pre-trained EQA model M
candidates ← Array[]
$A \leftarrow M.predict(P, Q)$
for $i = 1$ *to* n **do**
  score$_i$ ← Gestalt Pattern Matching(A, C_i)
  if $score_i \geq t$ **then**
    candidates.append((i, score$_i$))
sort candidates descending by score
return first k elements of candidates

2.2 Learning Methods

The candidates $\mathcal{T}$ selected in the last subsection are used as the ground truth to train an MCQA model. Because the candidates are not always correct, and each question can have multiple choices selected in the candidate set, the typical supervised learning approaches cannot be directly applied here. Therefore, the following learning methods are explored to form our objective function $\mathcal{L}$ for training the MCQA model from the candidates.

Highest-Only

$$\mathcal{L} = -\log P\left(c_{max} \mid p; q\right),$$

where c_{max} is the choice of a question q in the candidate set with the highest score. The approach here has no difference from typical supervised learning, except that the ground truth is from the candidate selection approaches, not human labeling.

Maximum Marginal Likelihood (MML)

$$\mathcal{L} = -\log \sum_{c_i \in \mathcal{T}} P\left(c_i \mid p; q\right)$$

In this objective, all the choices in the candidate set are considered as correct. The learning target of the MCQA model is to maximize the probabilities that all the choices in the candidate set labeled as correct. If there are more correct choices than the incorrect ones in the candidate set, the impact of the wrong choices in the candidate set can be mitigated.

Hard-EM Proposed by Min et al. (2019), this can be viewed as a variant of MML,

$$\mathcal{L} = -\log \max_{c_i \in \mathcal{T}} P\left(c_i \mid p; q\right)$$

The underlying assumption of this objective can be understood as follows. For a question q, several choices are selected in the candidate set. Although we don't know which one is correct, we assume one of them is correct. Therefore, we want the MCQA model to learn to maximize the probability of one of the choices for a question.

3 Experiments Setup

To evaluate the effectiveness of the proposed method compared to supervised learning and other approaches which do not require training data, we experiment on two MCQA tasks, RACE and MCTest(MC500).

3.1 Datasets

RACE Lai et al. (2017b) introduced the RACE dataset, collected from the English exams for middle and high school Chinese students. RACE consists of near 28000 passages and nearly 100000 questions. Specifically, the dataset can be split into two parts: RACE-M, collected from English examinations designed for middle school students; and RACE-H, collected from English examinations designed for high students. RACE-H is more difficult than RACE; and the length of the passages and the vocabulary size in the RACE-H are much larger than that of the RACE-M.

MCTest Richardson (2013) present MCTest which requires machines to answer multiple-choice reading comprehension questions about fictional stories. MCTest has two variants: MC160, which contains 160 stories, and MC500, which contains 500 stories. Moreover, MC500 can be subdivided into MC500-One and MC500-Multi. MC500-One refers to the questions that can be answered with one sentence. MC500-Multi refers to the questions that need evidence in multiple sentences to answer.

The length of each story is approximately from 150 to 300, and the topic of a story is a wide range, including vacations, animals, school, cars, eating, gardening, fairy tales, spaceships, and cowboys. Furthermore, the questions are marked whether it requires one or multiple sentences to answer. In our experiment, we evaluate our model on MC500 since there are only 280 questions in the MC160, which is not suitable in our setting.

Appendix A shows more details about both datasets.

3.2 Model Description

In this work, we used BERT-base Devlin et al. (2019) as the pre-trained model for both EQA system and MCQA system in the following experiments.

EQA model The hyperparameters we used are the same as official Released for training SQuAD 1.1. For both datasets, the EQA model is pretrained on SQuAD 1.1.

MCQA Model Similar to BERT finetuning on SWAG dataset (Zellers et al., 2018), we construct four input sequences, each containing the concatenation of the passage, the question, and one of the choices. The separator tokens `[SEP]` is added between the passage and the question. Next, we fed

	RACE		RACE-M		RACE-H		MC500		MC500-One		MC500-Multi.	
	dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
SW Matching Algorithm												
SW	30.8	30.2	36.2	35.2	28.4	28.1	46.5	42.8	36.7	43.7	54.5	42,1
Highest-Only	31.8	30.8	37.5	36.4	29.4	28.5	46.0	42.3	44.4	41.5	47.2	43.0
MML	34.0	33.1	40.3	40.5	31.4	30.1	50.0	45.3	**46.6**	44.4	52.7	**46.1**
Hard-EM	**34.3**	**34.0**	**41.0**	**41.2**	**31.5**	**31.0**	**51.5**	**45.7**	44.4	**47.7**	**57.3**	44.0
EQA Matching Algorithm												
EQA Match	32.3	32.2	40.3	40.5	28.9	28,8	62.5	**64.1**	**75.6**	**80.9**	51.8	49.8
Highest-Only	37.0	36.9	48.8	46.1	32.1	33.1	**67.5**	60.6	67.7	66.0	**67.2**	56.0
MML	38.6	**39.4**	**49.7**	49.6	34.0	**35.2**	65.5	61.3	67.8	67.1	63.6	56.3
Hard-EM	**39.1**	39.2	49.0	**49.7**	**35.0**	34.9	66.0	63.3	68.9	66.0	63.6	**60.9**
Supevised	64.9	65.5	70.0	71.0	64.0	63.3	70.0	64.3	75.6	69.0	60.4	65.4

Table 1: **Results on RACE and MC500 of MCTest.** The Supervised Learning was training with ground truth and used the same hyperparamter as others.

	RACE		MC500	
	dev	test	dev	test
SW Matching Algorithm				
Avg. numbers of candidates	3	3	1.98	1.85
Percent Including Answer	79.2	79.0	67.0	62.1
EQA Matching Algorithm				
Avg. numbers of candidates	1.35	1.38	1.63	1.62
Percent Including Answer	40.9	41.8	73.0	71.5

Table 2: **Number of Candidate Sets chosen by EQA and SW Matching.** *Percent Including Answer* means the percent of candidate set including the labeled answer.

	RACE-train	MC500-train
Case1	29759	202
Case2	8461	194

Table 3: **Candidate Set Analysis of RACE and MC500 of MCTest.** *Case1: candidates chosen by EQA including the answer but candidates chosen by SW not including the answer. Case2: candidates chosen by SW including the answer but candidates chosen by EQA not including the answer.*

the [CLS] token representation to the classifier and got the scores for each choice.

3.3 Training Details

We finetuned all models with a linear learning rate decay schedule with 1000 warm-up steps. The batch size is 32, and the max length of input size is 320. For RACE, we set the threshold to 0, max number of candidates to 3 with SW Matching, and set the threshold to 50, max number of candidates to 3 with EQA Matching. For MC500, we set the threshold to 3, max number of candidates to 2 with SW Matching, and set the threshold to 50, max number of candidates to 3 with EQA Matching.

Following Min et al. (2019), when we use hard-EM as objective, we perform annealing: at training step t, the model use MML as objective with a probability of $\min(t/\tau, 0.8)$ and otherwise use hard-EM, where τ is a hyperparameter. We tried $\tau = 1000$, 4000, and 8000.

4 Experiment Results & Analysis

4.1 Results

Table 1 shows the results of baselines and our methods on RACE and MC500.

RACE We found that our methods outperform SW and EQA Match across all the datasets with absolute gain 4$\sim$9%, which shows the MCQA model can improve itself from the noisy candidate sets. MML and Hard-EM outperform Highest-Only in all cases, which indicate that only relying on the single choice with the highest score is not sufficient. The improvement with EQA Mathcing Algorithm is more significant than with SW Matching Algorithm. This implies Candidates Choosing stage plays a significant role in the performance; more details will be discussed later.

MC500 With the SW Matching algorithm, our methods outperform performance baseline across all the datasets with absolute gains of 1$\sim$5%. With the EQA Matching Algorithm, although the performance of our method drops in MC500-One, we improve the MC500-Multiple performance by about 12%. This is because EQA models and SW can better capture the information within a sentence than multiple sentences, leading MC500-One performance much better than MC500-Multi with

EQA models and SW. This also shows that our method is not just picking the choice which has the most overlapped between context and the choice. Noticed that we achieved a comparable result to supervised learning on MC500 test-set.

4.2 Analysis

Candidate Set We found that the quality of the EQA matching method is much better than the quality of the SW matching method. Furthermore, our learning approach can almost dig out the correct answer from the candidate sets with EQA matching methods, which is one of the clues that our methods do work. Another evidence showing that our learning strategy truly learns something beyond the candidate selection approach is that the models' performance is much higher than the reference = (Percent Including Answer) / (Avg. numbers of candidates). This implies that after training, the MCQA models do not just randomly choose a prediction from the candidates. Noticed that "Average number of candidates" in SW is much larger than in EQA. This is the reason why "Percent Including Answer" results are much better for SW on RACE than EQA and we choose to analysis the reference instead of "Percent Including Answer". The number of "Percent Including Answer" and "Avg. numbers of candidates" are listed in Table 2.

Matching Methods For SW candidates, we only consider overlapped word between (question, choice) and context, but did not consider semantics of questions. On the other hand, EQA model considers the semantics of questions and extracts the answer from documents without using choices. The candidates are chosen by choice and extracted answer similarity. This makes the EQA candidates with higher confidence than SW candidates. We did the quantitative analysis to comapare two methods and show the result in Table 3. We consider the examples in two cases; case1: candidates chosen by EQA including the answer but candidates chosen by SW not including the answer; case2: candidates chosen by SW including the answer but candidates chosen by EQA not including the answer. Although the average number of candidates by SW is larger than by EQA (Table 2), there are more case1 examples in the training-set of both datasets, showing candidates picked by EQA have higher quality than candidates picked by SW. Note that we do not use the candidates set when testing.

5 Conclusion

In this paper, we proposed an unsupervised MCQA method, which exploits the pseudo labels generated by some basic rules or external non-MCQA dataset. The proposed method is shown to significantly outperform the baseline approaches on RACE and even comparable with the supervised learning performance on MC500. We hope this paper sheds light on zero-shot learning in NLP tasks.

References

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Unifying question answering, text classification, and regression via span extraction.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017a. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017b. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *EMNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

John W. Ratcliff and David Metzener. July 1988. Pattern matching: The gestalt approach.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Matthew Richardson. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Emprical Methods in Natural Language Processing (EMNLP 2013)*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: Aset of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference.

A Dataset Details

	RACE		
	train	dev	test
RACE-M	25421	1436	1436
RACE-H	62445	3451	3698

	MC500		
	train	dev	test
MC500-One	564	90	277
MC500-Multi	636	119	323

Table 4: **Number of examples in RACE and MC500 of MCTest.** RACE-M and MC500-One are easier than RACE-H and MC500-Multi separately.