RESEARCH-ARTICLE

Answer Selection Using Reinforcement Learning for Complex Question Answering on the Open Domain

ANGEL FELIPE MAGNOSSAO DE PAULA, University of São Paulo, Sao Paulo, SP, Brazil

ROBERTO FRAY DA SILVA, University of São Paulo, Sao Paulo, SP, Brazil

BRUNO EIDI NISHIMOTO, University of São Paulo, Sao Paulo, SP, Brazil

CARLOS EDUARDO CUGNASCA, University of São Paulo, Sao Paulo, SP, Brazil

ANNA HELENA REALI COSTA, University of São Paulo, Sao Paulo, SP, Brazil

Published: 19 February 2021

ISEEIE 2021: 2021 International Symposium on Electrical, Electronics and Information Engineering
February 19 - 21, 2021
Seoul, Republic of Korea

https://doi.org/10.1145/3459104.3459149
ISBN: 9781450389839

Answer Selection Using Reinforcement Learning for Complex Question Answering on the Open Domain

Angel Felipe Magnossão de Paula
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
angel.magnossao@gmail.com

Roberto Fray da Silva
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
roberto.fray.silva@gmail.com

Bruno Eidi Nishimoto
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
brunoeidinishimoto@gmail.com

Carlos Eduardo Cugnasca
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
carlos.cugnasca@usp.br

Anna Helena Reali Costa
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil
anna.reali@usp.br

ABSTRACT

Multiple-choice question answering for the open domain is a task that consists of answering challenging questions from multiple domains, without direct pieces of evidence in the text corpora. The main application of multiple-choice question answering is self-tutoring. We propose the Multiple-Choice Reinforcement Learner (MCRL) model, which uses a policy gradient algorithm in a partially observable Markov decision process to reformulate question-answer pairs in order to find new pieces of evidence to support each answer choice. Its inputs are the question and the answer choices. MCRL learns to generate queries that improve the evidence found for each answer choice, using iteration cycles. After a predefined number of iteration cycles, MCRL provides the best answer choice and the text passages that support it. We use accuracy and mean reward per episode to conduct an in-depth hyperparameter analysis of the number of iteration cycles, reward function design, and weight of the pieces of evidence found in each iteration cycle on the final answer choice. The MCRL model with the best performance reached an accuracy of 0.346, a value higher than naive, random, and the traditional end-to-end deep learning QA models. We conclude with recommendations for future developments of the model, which can be adapted for different languages using text corpora and word embedding models for each language.

CCS CONCEPTS

• **Information systems: Question answering; Query reformulation**; • **Computing methodologies: Information extraction; Natural language generation; Natural language processing; Partially-observable Markov decision processes; Neural networks**; • **Theory of computation: Reinforcement learning**;

ISEEIE 2021, February 19–21, 2021, Seoul, Republic of Korea

ACM ISBN 978-1-4503-8983-9/21/02…$15.00
https://doi.org/10.1145/3459104.3459149

KEYWORDS

Multiple-Choice Question Answer, Deep Reinforcement Learning, Deep Q-Learning

ACM Reference Format:
Angel Felipe Magnossão de Paula, Roberto Fray da Silva, Bruno Eidi Nishimoto, Carlos Eduardo Cugnasca, and Anna Helena Reali Costa. 2021. Answer Selection Using Reinforcement Learning for Complex Question Answering on the Open Domain. In *2021 International Symposium on Electrical, Electronics and Information Engineering (ISEEIE 2021), February 19–21, 2021, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3459104.3459149

1 INTRODUCTION

Natural language processing can be characterized as a group of techniques and methods that focus on processing and extracting information from natural language, free form texts [1]. In this paper, we focus on applying methods with deep reinforcement learning (DRL) for the task of answering complex multiple-choice questions on the open domain. This task involves a subset of the question answering (QA) problems in which there are multiple possible choices for answers. The objective is to evaluate the pieces of evidence that support each possible answer and select the most suitable one. Important examples for models that try to solve it are [2] and [3].

The main application of this task is self-tutoring, in which a student evaluates and learns from unlabeled multiple-choice questions without a teacher to point out the correct answer and the pieces of evidence that support it. A QA model must be able to provide the most likely choice and the text passages that support it.

The main difficulties in this task are: (i) identifying pieces of evidence that support each answer choice in the text corpora [3]; (ii) a need to understand long questions [4]; and (iii) that the answer may require multiple passages of text [4].

The traditional models used for solving QA problems, such as information retrieval (IR) and machine reading comprehension (MRC), do not address this problem satisfactorily [4]. The state-of-the-art model for this task [3] consists of an essential terms selector (ETS), which extracts terms from the question-answer choice pair; an IR model to select the text passages that support each choice; and an MRC model to evaluate the choices and find the best answer.

The answer chosen is the one that has the best pieces of evidence, and that contains the essential terms [2].

Nevertheless, three main problems need to be addressed to improve the results on this task: (i) lack of pieces of evidence found by the IR in the case of long and complex questions; (ii) difficulty of training the model due to the lack of labeled multiple-choice QA datasets; and (iii) difficulty of using the model on different languages. In this paper, we will implement deep reinforcement learning (DRL) to address those three problems, by reformulating the question-answer pairs when satisfactory pieces of evidence are not found [5] [3].

DRL is little explored in the literature for this problem. Its main advantage is allowing model training with unlabeled datasets and fewer data points [5]. This group of techniques addresses the current problem of lack of high quality, labeled datasets for this task. The results in the literature are promising using this method.

We propose the Multiple-Choice Reinforcement Learner (MCRL) model, which has components inspired in the current models (ETS, IR, and MRC) and a DRL with a policy gradient algorithm. Its focus is on extracting relevant pieces of evidence from several passages of text and reformulating the question-answer pairs. We then conduct an in-depth evaluation of the most critical hyperparameters of the model: reward function design, the weight of the pieces of evidence, and the number of iteration cycles. We used the ARC dataset [6] and accuracy and mean rewards per episode as quality metrics.

The paper is organized as follows: Section 2 contains fundamental concepts; Section 3 formulates the problem; Section 4 describes the MCRL model; Section 5 describes experiments and their results; Section 6 presents related work; Section 7 discusses the model's main impacts; and Section 8 concludes the paper.

2 FUNDAMENTALS

This section contains the relevant basic concepts related to possible solutions for the multiple-choice open-domain QA task, the use of DRL, the framework of Partially Observable Markov Decision Process (POMDP).

According to the literature, there are several possible options for solving this task, each with advantages and disadvantages:
(1) **Using only an IR model**: This model can quickly find pieces of evidence for the answer choices, but is not able to extract and interpret multiple text passages, leading to poor results;
(2) **Using a model with IR and MRC**: It can answer simple questions, but presents problems when dealing with complex questions. It is not able to reformulate queries to find new pieces of evidence when the current ones are not enough;
(3) **Using the model by [3], which focuses on questions with multiple choices, by combining IR, MRC, and ETS**: Although this model is better than options 1 and 2, it does not provide good results when the pieces of evidence found for the answer choices are not satisfactory. On the current paper, reformulating the queries will be evaluated as an alternative;
(4) **Using the model by [3] with an additional Reinforcement Learning (RL) component for generating new queries for each choice, using several iteration cycles:** This is the model proposed in this paper. The model uses several iteration cycles to generate new queries for each answer choice and provide new pieces of evidence. The model saves the score of each choice after each iteration cycle. Then, the MRC will consider a final ranking at the end of all the cycles, based on a mean of the choice probabilities on each cycle.

The DRL framework is used to train autonomous agents to learn through interactions with the environment, by performing actions, evaluating state changes, and receiving rewards or punishments depending on the quality of the action taken [7]. This quality of action involves predicting the expected future rewards by choosing an individual action at a specific state. The policy determines the quality of the actions and guides the actions taken.

Markov Decision Process (MDP) is used to model many sequential decision problems [8] solved by RL. MDP is used when the agent's decision depends directly on the last action taken. However, in some situations, the DRL agent can only observe part of the environment, or it may receive a considerable amount of noise on its inputs. In those cases, one can model the problem using a Partially Observable Markov Decision Process (POMDP) [9].

A POMDP is an abstraction of an MDP in which the agent has the same system dynamics as an MDP agent [9] but can only observe part of the state. As a result, the agent must identify and update a probability distribution considering the set of possible states.

A POMDP is described by the tuple $\langle S, A, T, R, O, \Omega \rangle$, where S is the set of environment **states**, A is the set of available **actions**, T is the **transition function**, R is the **reward function**, O is a set of **observations**, and Ω is an **observation function**. The agent does not know T and R so it must estimate and constantly update their values. The agent's goal is to learn an optimal policy π^* that maps each observation to the actions that lead to the highest expected cumulative sum of rewards over the agent's lifetime.

3 PROBLEM FORMULATION

The task of complex QA on the open domain with multiple choices can be defined as the following sequence of steps [3]:

(1) The model receives a question Q and N answer choices C_n, with $\boldsymbol{C} = \{C_n\}_{n=1}^{N}$, and $C_x \in \boldsymbol{C}$ being the correct one;

(2) It formulates N queries H_n for the IR model, based both on the question Q and its answer choices $\boldsymbol{C}$, forming the set of queries $\boldsymbol{H} = \{H_n\}_{n=1}^{N}$;

(3) The IR model searches in the text corpora $T_{corpora}$ for the passages of text (pieces of evidence) P_n that better answer each query H_n, forming the set of pieces of evidence $\boldsymbol{P} = \{P_n\}_{n=1}^{N}$;

(4) The MRC model receives set of pieces of evidence $\boldsymbol{P}$ and chooses the best possible answer choice C_x for the question Q.

A successful model for the self-tutoring problem would allow a student to learn a subject without the need of a professor, understanding: (i) the correct answers for each question; and (ii) the pieces of evidence that support them. Therefore, our research question is the following: "*Can RL improve the solution for the multiple choices complex open-domain QA task?*".

4 MULTIPLE-CHOICE REINFORCEMENT LEARNER

Our model has the following assumptions: (i) only one choice must be correct; (ii) the question must refer to material up to the 9th grade; and (iii) the question is formulated using proper grammar.

As shown in Figure 1, the MCRL is comprised of the models: (i) Essential Term Selector (ETS); (ii) Question Generator 1 (QG1); (iii) Information Retriever (IR); (iv) Machine Reading Comprehension (MRC); (v) Question Generator 2 (QG2); and (vi) Passage Ranking (PR). We designed the ETS, QG1, IR, and MRC models based on [3], and the QG2 and PR models based on [4]. Based on [5], we incorporate the idea of iteration cycles, adding a new step for the model: reformulating the queries and repeating steps 2 to 4 until the agent finds a satisfactory answer choice.

The central role of ETS is to search for the most relevant terms in a question Q, related to the answer choices $\boldsymbol{C}$. Therefore, both the question Q and each answer choices $\boldsymbol{C}$ form the input for ETS, and the resulting output is the selected terms $\boldsymbol{E}$. The QG1 model then uses $\boldsymbol{E}$ and C to generate queries $\boldsymbol{H}$ on the IR. Its main objective is to find evidence that can support each answer choice C_n. Each evidence is a text passage contained in the text corpora $T_{corpora}$, and the sum of the pieces of evidence found at the end of the final iteration cycle will form the final pieces of evidence for the answer choice concerning the specific question.

IR is a model built to return a set of pieces of evidence $\boldsymbol{P}$ from the $T_{corpora}$divided in N subsets P_n in which each one supports one of the answer choice C_n, given one query H_n. The resulting pieces of evidence $\boldsymbol{P}$ are sent both to the MRC and PR models

At the end of each iteration cycle, the MRC model receives as input the set of pieces of evidence $\boldsymbol{P}$ from the IR and Q and $\boldsymbol{C}$. If the current cycle is the last one, it receives the ranking $\mathbf{R}$ from the PR model instead of the set of pieces of evidence $\boldsymbol{P}$. It uses those inputs at each iteration cycle to determine the matching scores $\mathbf{SC}$ and the state $\mathbf{o}$ of the MRC.

The state $\mathbf{o_t}$ of the MRC represents what the agent can see from the whole environment at the time step t. QG2 later uses this state as one of its inputs. The QG2 model takes the queries generated by QG1 (if this is the first cycle) or QG2 (if this is any other cycle) in the previous step $\boldsymbol{H}_{t-1}$ and the state of the MRC $\mathbf{o_t}$ to generate new queries $\boldsymbol{H}_t$. The model uses these new queries to find better pieces of evidence P_n to support each answer choice C_n.

During the set of iteration cycles, PR collects all the sets of pieces of evidence $\boldsymbol{P}$ for the answer choices $\boldsymbol{C}$ provided by IR. It ranks each evidence for each answer choice C_n based on how many times IR returned it on the iteration cycles. Pieces of evidence that appeared more times have a higher ranking. The PR output are the top k evidences $\mathbf{R}$ from $T_{corpora}$ to support the answer choices $\mathbf{C}$. We separated the model into two phases.

Phase 1

As shown in algorithm1 , ETS selects relevant words from the question $\boldsymbol{E} \subset Q$, and then QG1 constructs N queries H_n joining $\boldsymbol{E}$ with each C_n. Next, IR uses theses queries to score all the sentences in $T_{corpora}$ and retrieve the top k sentences P_n. The MRC model receives N triples $\{\mathrm{Q},\ C_n,\ P_n\}$ to generate the $\mathbf{o_t}$ and N matching scores $\boldsymbol{SC_t} = \{\mathrm{SC}_n\}_{n=1}^{N}$.

Phase 2

The agent is composed of the QG2 and IR models. Its objective is to find the best $\mathbf{P}$ to support $\mathbf{C}$. First,QG2 receives $\mathbf{o_t}$ and the previous queries $\mathrm{H_{t\text{-}1}}$ and then generate a new group of queries $\mathrm{H_t}$. Secondly, the IR model repeats the same process carried out in phase 1,applying $\mathrm{H_t}$ to score all the sentences in the $\mathrm{T_{corpora}}$ to get the top k piece of evidence that can support each $\mathrm{C_n}$. In the final cycle, MRC receives $\mathbf{R}$ from PR instead of $\mathbf{P}$ from IR and returns $\mathbf{o_t}$ and $\mathbf{SC_t}$. MCRL applies a softmax function on the $\mathbf{SC_t}$ to determine the correct answer $\mathrm{C_x}$.

Algorithm 1 MCRL

Inputs: Question text (Q), Text corpora ($\mathrm{T_{corpora}}$), Model (M), Number of cycles (T), Number of top ranked sentences (k), Answer choices ($\boldsymbol{C}$)
Output: Selected Answer choice (C_x)
Phase 1
E ← M. essencial_term_selector(Q, $\boldsymbol{C}$)
$\mathrm{H_0}$ ← *M.query_generator1*(E, $\boldsymbol{C}$)
$\mathrm{P_0}$ ← *M. information_retriver*($H_0, T_{corpora}$, k)
$\mathrm{R_t}$ ← *M. passage_ranking*(p_0)
$\mathrm{SC_0}$,$\mathrm{o_0}$ ← *machine_reading_comprehension*(Q, $\boldsymbol{C}$, P_0)
Phase 2
for *t in range* (T) **do**
$\mathrm{H_t}$ ← *M.query_generator2*(o_t, Q)
$\mathrm{P_t}$ ← *M. information_retriver*($H_t, T_{corpora}$, k)
$\mathrm{R_t}$ ← *M. passage_ranking*(p_t)
if $t < T$ **then**
$\mathrm{SC_t}$,$\mathrm{o_t}$ = *machine_reading_comprehension*(Q, $\boldsymbol{C}$, P_t)
else
$\mathrm{SC_t}$,$\mathrm{o_t}$ = *machine_reading_comprehension*(Q, $\boldsymbol{C}$, R_t)
$C_x = Softmax(SC_t)$
end if
end for
return C_x

During the MCRL learning process, the POMDP is the following:
S: Composed of the question, answer choices, essential terms, text corpora, and queries.
O: Given by the state of machine reading comprehension.
A: Given by all phrases in the text corpora.
T: The environment progresses deterministically after reading the paragraphs sent by the information retriever model.

The parameters in the QG2 model represent the policy π. The QG2 model is composed of a Gated Recurrent Unit and a Feed-Forward Neural network. We use a policy gradient algorithm to guide the construction of the agent's policy, as in [4].

The r_t represents the reward at time t returned by the environment after each action by the agent. The T represents the number of interactions cycles. To maximize the expected reward, the agent optimizes the QG2 parameters:

$$J(\theta) = E_\pi \sum_{t=1}^{T} r_t \tag{1}$$

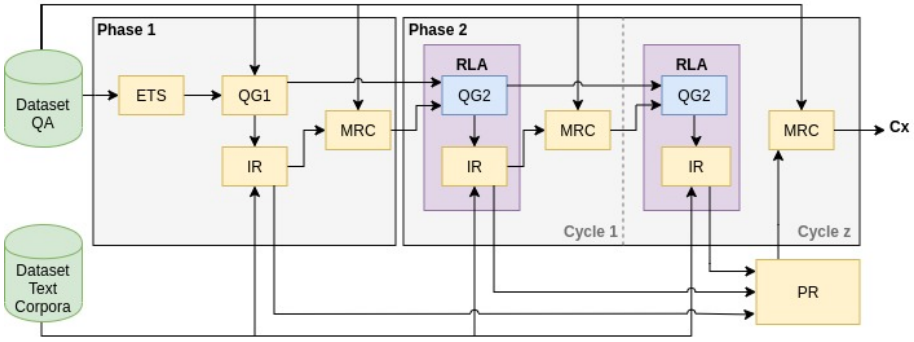

Figure 1: Main components of the MCRL model.

We calculate the gradient using the REINFORCE algorithm [10]:

$$\nabla_\theta J(\theta) = E_\pi \sum_{t=1}^{T} r_t \cdot \log\left(\pi_\theta\left(p_t|q\right)\right)) \tag{2}$$

Let q denote the query and p_t the top k sentences sent to the MRC model at time t. We apply a softmax function to compute the probability that the current policy assigns to p_t, as in [4],

$$\pi_\theta\left(p_t|q\right) = softmax\left(score\left(p_t|q\right)\right) \tag{3}$$

5 EXPERIMENTS

This section contains a description of the experiments conducted to evaluate the proposed model. The dataset used was the ARC dataset [6]. The ARC dataset contains 7.787 natural science questions of varying complexity, with up to 5 possible answer choices, extracted from exams for students from 3rd to 9th-grade [6]. The length of each question varies from 3 to 128 words, with an average of around 20. The vocabulary size contains 6.329 words after stemming [6].

A simple query on an IR model cannot answer most of the questions in this dataset. Its main differences in comparison with other QA datasets are: (i) it has easy and challenging questions; (ii) the majority of questions demand reasoning or interpretation; (iii) all questions have multiple answer choices; and (iv) the questions encompass various categories. We carried out our experiments in the challenging set, which is composed of train, validation, and test subsets, consisting of 1.119, 299, and 1.172 questions.

We used the following Python libraries: NumPy, Pandas, Scikit learn, PyTorch, TensorFlow, Matplotlib, and SciPy. The techniques used for preprocessing were tokenization, stemming, part-of-speech, stop-words-removal, and named-entity recognition [3].

The first experiment was exploring different weights of the components of the reward function. We evaluated two possibilities: (i) equal weights for the differences between the score in time t and t - 1 for all possible answer choices, as shown in equation:

$$r_1 = \sum_{i}^{N} \left(SC_{i_t} - SC_{i_{t-1}}\right) \tag{4}$$

and (ii) considering double weight for the difference between the score in time t and t-1 for the right answer choice and keeping simple weight for the rest of the differences:

Table 1: Models implemented.

Model	Reward Function	Number of cycles	Information weight
M1	1	1	1,2
M2	1	3	1,2
M3	1	5	1,2
M4	2	1	1,2
M5	2	3	1,2
M6	2	5	1,2

$$r_2 = 2(SC_{r_t} - SC_{r_{t-1}}) + \sum_{i}^{N-1} (SC_{i_t} - SC_{i_{t-1}}) \tag{5}$$

The second experiment was related to how the information of each iteration cycle was used on the final prediction, considering: (i) equal weights for all the cycles:

$$W_{Passage\ Ranking\ 1} = 1 \tag{6}$$

and (ii) the first cycle as having double the weight of the other cycles on the final prediction:

$$W_{Passage\ Ranking\ 2} = \begin{cases} if\ cycle = 1 \ \rightarrow 2 \\ else \ \rightarrow 1 \end{cases} \tag{7}$$

The central hypothesis behind this formulation is that, as more iteration cycles happen, the query starts to deviate from the original question-answer pair. For this reason, we expect that the first reformulation (first iteration cycle) should gather the most information necessary to answer the question.

The third experiment was related to the different number of iteration cycles. The primary rationale behind it is that there will be an optimal number of iteration cycles for the RL agent, as observed by [5]. We considered the following options: 1, 3, and 5 cycles.

Table 1 describes the models evaluated and their hyperparameters. Table 2 contains the accuracy and mean rewards per cycle of all the experiments. All the experiments considered convergence on 10.000 episodes. More cycles led to higher running time.

We can conclude that: (i) the reward function 1 provided the best results in terms of accuracy (0.338 vs. 0.335 for the reward function 2); (ii) the models with 1 iteration cycle presented the best accuracy

Table 2: Results of the hyperparameter analysis of the model on the validation subset. The best results are in bold.

Model	Information weight	Accuracy	Mean rewards/cycle
M1	1	**0.359**	4.123
	2	**0.359**	4.124
M2	1	0.336	1.764
	2	**0.359**	1.809
M3	1	0.322	1.311
	2	0.295	1.311
M4	1	0.356	**9.915**
	2	0.356	**9.915**
M5	1	0.326	0.847
	2	0.356	0.847
M6	1	0.299	0.791
	2	0.319	0.759

Table 3: Results of the final models.

Model	Accuracy	Mean rewards/cycle	Total time (h)
M1	0.343	4.628	**7.96**
M2	**0.346**	1.995	15.18
M3	0.306	1.385	21.74
M4	0.345	**10.702**	**7.96**
M5	0.328	0.979	15.18
M6	0.311	0.826	21.74

(0.357 for 1 cycle vs. 0.344 for 3 cycles and 0.309 for 5 cycles); and (iii) considering the weight of the first cycle the double of the weight of the other cycles provided the best accuracy (0.341 vs. 0.333 for considering the same weight for all cycles). M1 and M2 provided the best results, with an accuracy of 0.359. The mean of rewards per cycle of M1 is more than twice as higher than that of M2, and more than three times higher than that of M3. It is important to note that M4 also obtained similar results in terms of accuracy.

M6 was the worst model, with an accuracy of 0.299. Its mean of rewards per cycle was also the lowest for the reward function 2, indicating that it did not capture as much information as the others for this reward function. Its mean of rewards per cycle was around 10% lower than M5, and more than ten times lower than M4.

Those results are in line with our initial hypotheses: (i) the model gathers most of the information on the first cycle; and (ii) the reward function should provide a higher weight for the first cycle. Nevertheless, more experiments are needed on different datasets.

Table 3 contains the results of the final models on the test subset. We evaluated the following metrics: accuracy, mean rewards per cycle, and total time in hours. The best model was M2, with an accuracy of 0.346. Nevertheless, both the models M4 and M1 had similar results. More experiments are needed with additional datasets to be able to conclude which of these models present the best accuracy. Based on the hyperparameter analysis, we infer that a model with 1 iteration cycle will present the best results.

Table 4: Comparison with state-of-the-art models on the test subset.

Model	Accuracy
IR solver [3]	0.203
Random [3]	0.250
BiDAF [3]	0.265
BiLSTM Max-out [3]	0.339
MCRL - M4 (our model)	0.345
MCRL - M2 (our model)	0.346
ET-RR (Concat) [3]	0.353
ET-RR [3]	**0.366**

For mean rewards per cycle, M4 presented the best results, with 10.702, followed by M1, with 4.628. These results are another indication that the models with 1 iteration cycle may present better results. The model with the worst mean reward per cycle was M6, with a value of 0.826. These results reinforce that having more cycles may not improve considerably on the information gathered.

While an artificial neural network may take seconds up to minutes to run on a test subset, an RL agent can take hours up to days. In the case of the MCRL, running on the test subset took about 8 hours for the models with 1 cycle, up to around 22 hours for the models with 5 cycles. These results show a significant increase in running time when increasing the number of cycles.

Table 4 contains the results of the comparison of the MCRL models with several other models [3]. The MCRL model (with 1 or 3 iteration cycles) provides better results than random, naive, and end-to-end deep learning models. Nevertheless, it did not provide better results than the state-of-the-art ET-RR model.

Nevertheless, using an RL agent has benefits concerning the ET-RR: (i) model training with unlabeled datasets and fewer data points; and (ii) the possibility of using on different datasets, as the reward function is not specific for a restricted domain.

We conclude that using RL to reformulate questions and evaluate pieces of evidence of question-answer pairs provide a satisfactory solution for the multiple-choice complex open-domain QA task.

6 RELATED WORKS

The first works on open-domain QA are from the 1960s [11]. The Trec-8 task was created to promote the area, launching many datasets [12–14]. The ARC dataset [6] is important for multiple-choice QA. Its questions cannot be answered by a model only based on the IR or Pointwise Mutual Information paradigm. The state of the art model scores a little better than a random guess [3].

Two developments have been suggested in recent years: (i) query reformulation; and (ii) use of iteration cycles. Both aim to improve the possibility of finding the correct answer. Query reformulation has been demonstrated to improve the IR results, by improving the original query [15]. This is shown by [16], which used RL to reformulate the queries.

The work of [5] has shown improvements over the results of the BiDAF reader model. The Multi-step Retriever-Reader developed by [4] is the closest to our work. Nevertheless, their model is not optimized for multiple-choice QA problems.

The use of iteration cycles has improved results for many QA tasks. The iteration cycles in [4] act as a type of controller update. Our work, based on [4], uses cycles to reshape queries and improve the possibility of finding the right answer.

7 DISCUSSIONS

An analysis of several instances of the model's results indicates that it is possible to group the text passages found by the model for each answer choice in three main categories: (a) passages that directly point to the correct answer choice; (b) passages that are only partially related to the correct answer choice; and (c) passages that are not related to any of the answer choices.

In Example 1, the model chose the wrong alternative. We believe that the main reason is that it did not capture what was being asked. The question asked was which of the answer options did not cause the evolution, but the model seems to have interpreted which was the leading cause of it. For this reason, it picked choice A. The pieces of evidence found for choice B were not as strong.

We can observe that: (i) the model is not able to capture subtle changes in the question phrasing; and (ii) the pieces of evidence gathered when the model does not understand the question are noise. Future work will address both of these points, by improving the number of questions with a negative connotation in the dataset and by further exploring how to better separate noisy passages, reducing the probability of choosing the wrong answer.

Example 1

Question: Biological evolution can occur through all of these except?

(A) competition

(B) fossilization ✓

(C) variation

(D) adaptation

There are two main explanations for this observation: (i) a small dataset, resulting in sub-optimal policies; and (ii) a lack of a clear metric for defining if the agent should go through an additional cycle. The first point could be addressed by using larger datasets. The second, by using a metric such as information gain to evaluate and help the agent define if more cycles are necessary.

8 CONCLUSIONS AND FUTURE WORKS

Improving the accuracy of multiple-choice QA for complex open domain questions could have a series of benefits. The main difficulties of this task are: (i) identifying pieces of evidence that support each answer choice; (ii) understanding long questions; and (iii) choosing the answer based on multiple passages of text.

We proposed the use of a deep RL model, MCRL, with several iteration cycles to reformulate queries for the IR component and then to choose the answer that has the highest probability of being the correct one. Although the proposed model did not achieve the same results as the state-of-the-art model, it was considerably close and still has room for improvement in several aspects.

The main limitations of our work are: (i) lack of available datasets for multiple-choice QA; and (ii) difficulty assessing the number of iteration cycles. Future works are related to: (i) improving the passage ranking algorithm, improving the information gathered through the cycles; (ii) using different datasets to train the agent; and (iii) using an unsupervised machine learning model to cluster the questions and answer choices before using the model.

ACKNOWLEDGMENTS

The authors acknowledge the support of CNPq under grants 425860/2016-7 and 307027/2017-1. This research is being carried out with the support of *Itaú Unibanco S.A.*, through the scholarship program of *Programa de Bolsas Itaú* (PBI), and it is also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Finance Code 001, Brazil.

REFERENCES

[1] Ruslan Mitkov. *The Oxford Handbook of Computational Linguistics (Oxford Handbooks)*. Oxford University Press, Inc., USA, 2005.

[2] Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue-Nkoutche, Pavan Kapanipathi, Nicholas Mattei, Ryan Musa, Kartik Talamadupula, and Michael Witbrock. A systematic classification of knowledge, reasoning, and context within the ARC dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 60–70, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[3] Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. 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 335–344, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. In 7th International Conference on Learning Representations, *ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. Ask the right questions: Active question reformulation with reinforcement learning. In *International Conference on Learning Representations*, 2018.

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[7] Stuart Russell and Peter Norvig. Artificial Intelligence: *A Modern Approach*. Prentice Hall, 3 edition, 2010.

[8] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 04-2011 edition, 2012.

[9] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[10] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992.

[11] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM Computer Conference*, pages 219–224, 1961.

[12] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

[13] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*, 2017.

[14] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[15] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR*, 1996.

[16] Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 574–583, Copenhagen, Denmark, September 2017. Association for Computational Linguistics