Teach Machines to Learn Main Content for Machine Reading Comprehension

Zhaohui Li, Yue Feng, Jun Xu, Jiafeng Guo, Yanyan Lan, Xueqi Cheng
CAS Key Lab of Network Data Science and Technology
Institute of Computing Technology, Chinese Academy of Sciences, China
{lizhaohui,fengyue}@software.ict.ac.cn, {junxu,lanyanyan,guojiafeng,cxq}@ict.ac.cn

Abstract

Machine reading comprehension (MRC) on real web data, which means finding answers from candidate passages according to a query, is an arduous task in the NLP community. One of the crucial problems is how to filter out the key information from the candidate passages in which contain a large amount of redundant and overlapping text. To tackle this issue, this paper defines a sub-task in MRC, which aims to select the **main content** from a set of passages based on the Query. And a particular end-to-end Markov-Decision-Process model (**MC-MDP**) to address this task. Specifically, an attention mechanism is employed to link and encode the query and passages and then a sentence sequence selection model is formulated by MDP and trained by Policy Gradient in Reinforcement Learning. Experiments are conducted on MSMARCO dataset and the result shows it can gain most part of the query information and can boost the following answer span prediction model by a large margin.

Introduction

Machine reading comprehension (MRC) aims to extract answers from a set of passages according to a query or question. In recent tasks, more and more candidate passages are involved in a query. So how to obtain answer-related information from the enormous texts can be a tough task in the MRC task. Many researches handled this problem by ranking the passage list or merge the answers derived from every passage, such as S-Net (Tan et al. 2017). These methods are not fully reasonable for two reasons: First, since many candidate passages have some similar content or even the wrong information, extracting answers from all of the passages means the overlapping information, non-related information and even the waste of calculating resource; Second, the ranking is not a valid method for MRC task, because MRC task is aimed to get a right answer from these passages, we just need to find the related texts which contain the right answer, and it is not necessary to rank the passages.

Inspired by real human Reading Comprehension process, that is, when we are doing an exam question, we may read the query, and then begin to read the passage to find the query-related content from start to end. Meanwhile, our attention is focused on the query-related text and ignores the irrelevant part. Then, we draw the final answer from these main content. This process can also be mimicked by Machines, that is, it can be formulated by a Markov Decision Process(MDP) (Puterman 2014) and trained by Policy Gradient method so as to make it mathematically computable. Also, given the popularity and effectivity of Attention Mechanism(Seo et al. 2016), a novel end-to-end Main Content (MC) selection model, which can boost the span predict model efficiently is implemented by this paper, which called MC-MDP. Compared to simple Golden Passage or Paragraph and the Ranking-based Method, this model can obtain a more exact main content, that is, higher F1 and Rouge-L scores. Experiments on MSMARCO dataset validate the effectiveness of this MC-MDP model. It should be pointed out that this model does not include the span predict part in MRC task, which is the last step and generates the final answer, and we just employ these state-of-the-art models to test the availability and efficiency of our MC-MDP model.

Query-based MC-MDP Model

This end-to-end model is composed of an encoding layer and an MDP sequence selection layer. The encoding layer employed Attention mechanism to obtain a question and passage representation for each data. And then an MDP model is used to generate the main content sequence.

Strictly following Bi-Direction Attention Mechanism(Seo et al. 2016), the first layer calculates the query-aware passage and passages-aware Question word representations. Then two BiLSTM are employed to encode the representation of question $\mathbf{Q}$ and each question' candidate sentences $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$, and M is the sentence number of passages of this question. The goal of the MDP layer is to construct an agent which can select a set of candidate sentences as Main Content. Thus, a special MDP is formulated as:

States $\mathcal{S}$: State are designed at step t as a triple $s_t = [\mathbf{Q}, \mathcal{Z}_t, X_t]$, where $\mathcal{Z}_t = \{\mathbf{x}_{(n)}\}_{n=1}^{t}$ is the sequence of t selected sentences, where $\mathbf{x}_{(n)}$ is the $\mathbf{t}^{th}$ sentence in this sequence; X_t is the set of candidate sentences. At the beginning ($t = 0$), the state is initialized as $s_0 = [\mathbf{q}, \emptyset, X]$, where $\emptyset$ is the empty sequence and X contains the top $\mathbf{w}$ sentences in all the candidate passages. Note that the agent read all candidate passage sentence from start to end, we design a w size window to regulate the agent's candidate Actions.

Actions $\mathcal{A}$: At each time step t, the $\mathcal{A}\left(s_t\right)$ is the set of actions the agent can choose, each corresponds to a sentence

Algorithm 1 MC-MDP

Input: Training set $D = \{(\mathbf{Q}^{(n)}, X^{(n)}, A^{(n)})\}_{n=1}^{N}$, learning rate η, dropout keep rate d, and value function R
Output: Parameters $\mathbf{\Theta}$

1: Initialize $\mathbf{\Theta} \leftarrow$ random values in $[-1, 1]$
2: **repeat**
3:   **for all** $(\mathbf{Q}, X, A) \in D$ **do**
4:     $SampleEpisode(\mathbf{s_0}, \mathbf{a_0}, \mathbf{r_1}, \cdots, \mathbf{s_{M-1}}, \mathbf{a_{M-1}}, \mathbf{r_M})$
5:     **for** $t = 0$ **to** $M - 1$ **do**
6:       $\mathbf{G_t} \leftarrow \sum_{k=0}^{M-1-t} \mathbf{r}_{t+k+1}$
7:       $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \eta \mathbf{G_t} \nabla_{\Theta} \log a_t | s_t; \Theta$
8:     **end for**
9:   **end for**
10: **until** converge
11: **return** $\mathbf{\Theta}$

from X_t. That is, the action $a_t \in \mathcal{A}(s_t)$ at the time step t selects a sentence $\mathbf{x}_{m(a_t)} \in X_t$ for the main content sequence, where $m(a_t)$ is the index of the sentence selected by a_t.

Transition T**:** The function $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined as two steps: 1) Append sentence $\mathbf{x}_{m(a_t)}$ to $\mathcal{Z}_t$; 2) Set X_{t+1} as $\{\mathbf{x}_{m(a_t)+1}, ..., \mathbf{x}_{m(a_t)+w}\}$, where w is the window size.

Reward R**:** The reward can be considered as the evaluation of the information quality of the main content sequence, and the reward function at t step is defined as the arithmetic mean of F1(t) and Rouge-L(t).

Policy function $\mathbf{p}$**:** The policy $\mathbf{p}(s)$ defines a function that takes the state as input and output a distribution over all of the possible actions $a \in \mathcal{A}(s)$. Specifically, each probability in the distribution is a normalized function whose input is the bilinear product of the LSTM and the selected sentence:

$$p(a|s) = \frac{\exp\left\{\mathbf{x}_{m(a)}^{T} \mathbf{U}_p \text{ LSTM}(s)\right\}}{\sum_{a' \in \mathcal{A}(s)} \exp\left\{\mathbf{x}_{m(a')}^{T} \mathbf{U}_p \text{ LSTM}(s)\right\}} \quad (1)$$

Learning with policy gradient The model parameters are donated as $\mathbf{\Theta}$. The training phase is showed in Algorithm 1. It uses Equation(1) to sample a episode $E = (\mathbf{s_0}, \mathbf{a_0}, \mathbf{r_1}, \cdots, \mathbf{s_{M-1}}, \mathbf{a_{M-1}}, \mathbf{r_M})$ to train the optimal policy: $\tilde{a} = \arg\max_{a \in \mathcal{A}(s)} \pi(a|s)$. The definition of long-term return G_t is crucially important, for it equals the ground truth here. So we define the discounted sum of rewards from position t as G_t, its detail can be found in Algorithm 1 line 6. After the training phase, the agent can select main content sentences from passages according to its policy function.

This Reinforcement learning method formulates the main content selection process as a sequence selection episode. By this method, answer information which is not in the same passage can be selected and the overlapping information can be filtered. Meanwhile, these two layers can be trained jointly. These are the merits of MC-MDP model.

Experiment and Analysis

Experiments are conducted on MSMARCO, a real-world dataset contains 100K queries and 1M passages, and its answers are generated by human beings. We use TextBlob to do the preprocess, and initiate word vectors by the 300-dimensional pre-trained GloVe embeddings. All hidden layers are 150-dimension and the dropout rate between layers is 0.8. Meanwhile, contrast experiments **GP**(Golden Passage) is conducted as well, which is a heuristic approach which chooses passage has the largest overlap with the question as the main content.

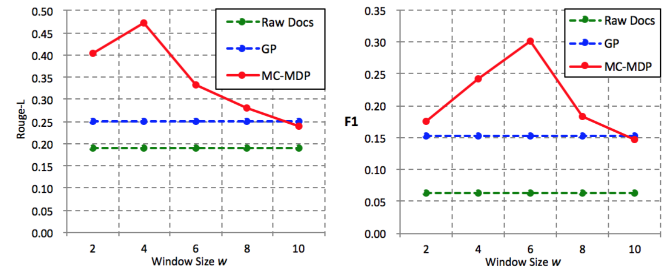

Figure 1: Scores of experiments with different methods

Table 1: Comparison of results in verification tests.

Method	ROUGE-L%	BLEU-4%
Match-LSTM	30.67	31.05
BiDAF	31.52	32.11
GP + Match-Lstm	37.33	40.72
GP + BiDAF	37.56	41.24
MC-MDP + Match-LSTM	**43.97**	**43.23**
MC-MDP + BiDAF	**44.12**	**43.45**

As it is shown in Figure 1, the main experiments' results which measured the main content selection ability of the model is evaluated by F1 and ROUGE-L. It makes sense that the medium window size can achieve the most robust policy in MDP model, and it can beat GP by a large margin.

In addition, the results of main experiments (w=5) are used as input in the verification tests in which we employ two state-of-the-art span prediction model **Match-LSTM** (Wang and Jiang 2016) and **BiDAF** to extract the final answer. We adopt the official evaluation metrics, including ROUGE-L and BLEU-1. And Table 1 shows the results of our model can improve the performance of MRC model.

Conclusion and Future Work

In conclusion, this paper defines a sub-task in multi-passage MRC, which is aimed at filtering out valuable information from massive texts according to the query. formulate as an end-to-end MDP model and it employs attention mechanism. Experiments show that it can filter out the query-based information and boost the state-of-the-art models. These prove that the MDP and RL method can be used in the MRC task, and it deserves more attention as well as researches.

References

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Tan, C.; Wei, F.; Yang, N.; Lv, W.; and Zhou, M. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *arXiv preprint arXiv:1706.04815*.

Wang, S., and Jiang, J. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.