Multi-choice Relational Reasoning for Machine Reading Comprehension

Wuya Chen[1], Chunyu Kit[2], Xiaojun Quan[1]*, Zhengcheng Min[1], Jiahai Wang[1]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou
[2]Department of Linguistics and Translation, City University of Hong Kong, Hong Kong
quanxj3@mail.sysu.edu.cn, ctckit@cityu.edu.hk

Abstract

This paper presents our study of cloze-style reading comprehension by imitating human reading comprehension, which normally involves tactical comparing and reasoning over candidates while choosing the best answer. We propose a multi-choice relational reasoning (McR^2) model with an aim to enable relational reasoning on candidates based on fusion representations of document, query and candidates. For the fusion representations, we develop an efficient encoding architecture by integrating the schemes of bidirectional attention flow, self-attention and document-gated query reading. Then, comparing and inferring over candidates are executed by a novel relational reasoning network. We conduct extensive experiments on four datasets derived from two public corpora, Children's Book Test and Who DiD What, to verify the validity and advantages of our model. The results show that it outperforms all baseline models significantly on the four benchmark datasets. The effectiveness of its key components is also validated by an ablation study.

1 Introduction

Machine reading comprehension (MRC) is a challenging task that requires much semantic understanding and reasoning using various clues from texts (Seo et al., 2016). Its general form is to ask a computer to answer questions in natural language according to its understanding of a given article or a context. As a specific form of MRC, cloze-style reading comprehension has recently gained increasing attention. Cloze-style MRC is a task to fill in a blank in the query with an appropriate word or phrase according to given context.Several large-scale datasets (Hill et al., 2016; Hermann et al., 2015; Onishi et al., 2016) for this task have been released, facilitating the development of various machine learning models (Kadlec et al., 2016; Trischler et al., 2016; Dhingra et al., 2017; Ghaeini et al., 2018). Most of these learning systems are built upon multi-hop architectures and attention mechanisms (Dhingra et al., 2017), which have been shown to excel at distilling useful information and learning the "importance" distribution over inputs. There are also some works that mimic the cognitive process of human reasoning with complicated hypothesis testing frameworks (Trischler et al., 2016; Munkhdalai and Yu, 2016). However, none of these models provide explicit reasoning among candidate answers with respect to a given context and query.As a practical skill for cloze test, humans often need to tactically compare two candidate answers while making decision, especially when more than one candidate appears to be competent.

In this paper, we propose a multi-choice relational reasoning (McR^2) model to imitate the above process. It first learns representations of document and query via a hierarchical multi-stage encoding architecture to explore the relations between document and query. The encoding architecture integrates the mechanisms of bidirectional attention flow (Seo et al., 2016), self-attention (Wang et al., 2017) and document-gated query reading (Dhingra et al., 2017) to learn dependencies between context and query and map them to representations rich in semantic information. The model then utilizes a multi-choice relational reasoning module to realize comparing and inferring over candidates.

*Xiaojun Quan is the corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 6448–6458
Barcelona, Spain (Online), December 8-13, 2020

Our reasoning module is inspired from the Relational Networks (Santoro et al., 2017) which have recently demonstrated a very success in several other relational reasoning tasks than MRC. To our knowledge, this work is the first attempt in cloze-style MRC to explicitly perform reasoning over candidate answers to facilitate answer deduction. We conducted extensive experiments on four datasets derived from two public corpora, Children's Book Test and Who DiD What, to verify the validity and advantages of our model. The experimental results confirm its outperformance over other state-of-the-art models. In addition, an ablation study also validates the effectiveness of its key components in contrast to other alternatives, and a case study with visualization further proves the effect of its relational reasoning module. Finally, by time analysis we show the advantage of our multi-stage encoding architecture for training.

2 Overview

The inputs of cloze-style reading comprehension can be represented as a tuple (D, Q, C, a), where $D = [w_1, ..., w_m]$ is a context document of length m, $Q = [q_1, .., q_n]$ represents a query of length n with a placeholder, and $C = [c_1, .., c_k]$ is a set of k candidates. As shown in Figure 1, the framework of our model can be divided into two parts, namely, a fusion representation module and a multi-choice relational reasoning module. While the first module aims to produce fusion representations for document and query by means of a hierarchical multi-stage architecture, the second module to perform comparing and reasoning over candidates based on the fusion representations. In the following sections, we first introduce the fusion representation, and then present the multi-choice relational reasoning module.

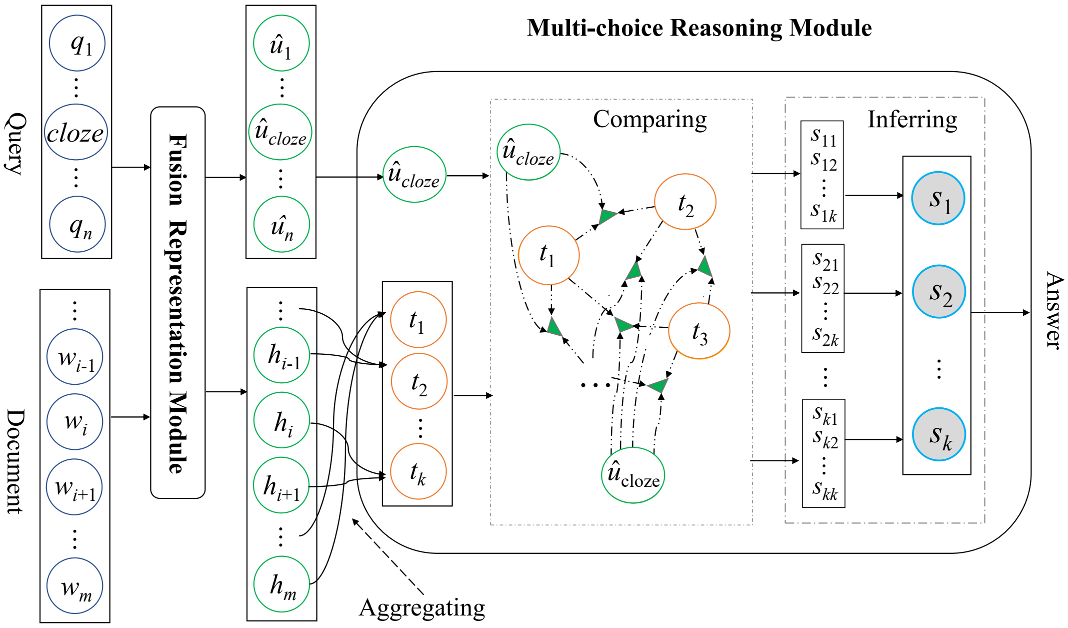


Figure 1: The framework of our McR^2 model, which can be divided into two parts, namely, a fusion representation module and a multi-choice relational reasoning module.

3 Fusion Representation

Taking into consideration a number of previous works (Seo et al., 2016; Wang et al., 2017; Dhingra et al., 2017; Kadlec et al., 2016), this module is designed to consist of four layer, which will be described in detail in the following paragraphs. It should be noted that this module is replaceable in our model, because our model is modular and applicable to many other encoding options, as we will discuss in the experiments section.

3.1 Contextual Embedding

In this layer, we first transform every word in document D and query Q into continuous vector representation with a shared pre-trained word embedding matrix. We then employ a bidirectional gated recurrent unit (Bi-GRU) to encode these word embeddings and obtain contextual vector representations $P \in R^{2d\times m}$ for a document and $U \in R^{2d\times n}$ for a query, where *d* is the output size of the Bi-GRU.

3.2 Bidirectional Matching

This layer aims to interconnect the contextual embeddings of query and document and produce a set of query-aware feature vectors for document words (Seo et al., 2016). Inputs of this layer include contextual vector representations $P \in R^{2d\times m}$ and $U \in R^{2d\times n}$ from the previous layer, and outputs are query-aware vector representations of document words. Specifically, for vector $p_i \in P$ and vector $u_j \in U$, corresponding to document word w_i and query word q_j respectively, the attention between w_i and q_j is computed as: $\alpha_{ij} = p_i \cdot u_j$. Then an attended vector o_i for each document word is computed as: $o_i = \sum_{j=1}^{n} u_j \mu_{ij}$, where

$$\mu_{ij} = \exp(\alpha_{ij}) / \sum_{k=1}^{n} \exp(\alpha_{ik}) \tag{1}$$

Next, a query-to-context vector $\widetilde{q_c}$ is obtained by: $\widetilde{q_c} = \sum_{i=1}^{m} p_i \lambda_i$, where

$$\eta_i = \max_{1<=j<=n} \alpha_{ij}, \tag{2}$$

$$\lambda_i = \exp(\eta_i) / \sum_{k=1}^{m} \exp(\eta_k), \tag{3}$$

Finally, query-aware vector $g_i \in G$ is computed for document word w_i by concatenating p_i, o_i, $p_i \odot o_i$ and $p_i \odot \widetilde{q_c}$, where $\odot$ denotes element-wise multiplication. In order to capture the interaction between document words with respect to query, we utilize a Bi-GRU to encode G and obtain a matrix $V \in R^{2d\times m}$ as coarse-grained fusion representation for document words.

3.3 Document Self-Matching

Due to the deficiency of RNNs, which bias current input within a window region and concerns little about other cues outside (Wang et al., 2017), the above representation may only contain finite knowledge of a document. In order to overcome this problem, we apply self-matching (Wang et al., 2017) on the document representation. The computation is similar to α_{ij} in the above but only between a document and itself, and the output is the fine-grained representation $H \in R^{2d\times m}$ for a document.

3.4 Document-Gated Query Reading

This layer computes document-specific query representation by means of a gated-attention mechanism (Dhingra et al., 2017). For a contextual vector u_i of a query, we calculate:

$$\delta_i = softmax(H^\top u_i), \tag{4}$$

$$u^* = H\delta_i, \tag{5}$$

$$\hat{u}_i = u_i \odot u_i^*. \tag{6}$$

where $\hat{u}_i \in \hat{U}$, and $\hat{U} \in R^{2d\times n}$ is the final representation for the query.

4 Multi-choice Relational Reasoning

This module mainly derives inspiration from the Relational Networks (Santoro et al., 2017) which have demonstrated its success in several relational reasoning tasks. The details of this module can be described in three steps: aggregating, comparing, and inferring.

4.1 Aggregating

Since a candidate answer may appear in multiple positions of a document, this step aims to generate a global representation for each candidate. Inspired by the pointer-sum attention mechanism Kadlec et al. (2016), we design a simple *pointer-sum vector* to obtain the global representation of a candidate by adding up its representations in different positions:

$$t_i = \sum_{j \in I(w,D)} h_j, \tag{7}$$

where w denotes a candidate answer, $I(w, D)$ is a set of positions where w appears in document D, and $h_j \in R^{2d}$ is the corresponding fine-grained vector representation of w in position j. This approach also benefits the cases when multiple words are included in a single candidate, where a pointer-sum vector will be generated based on the representations of these words. Finally, we obtain a matrix $T \in R^{2d \times k}$ of global representations for all candidates, where k is the number of candidate answers.

4.2 Comparing

Our basic assumption is that comparing among different candidates based on query plays a critical role in reading comprehension. To represent a query, we choose not to use its whole representation but only use its cloze representation $\hat{u}_{cloze} \in \hat{U}$, as we believe the cloze representation already encodes sufficient contextual information for this task. We then compare two candidates and query by concatenating their representations and passing the result through a neural network g_θ:

$$\mathcal{S}_{ij} = g_\theta([t_i; t_j; \hat{u}_{cloze}]), \tag{8}$$

where t_i and t_j are the representations of two candidates. $\mathcal{S}_{ij}$ is a vector that can be used to measure the "likelihood" of candidate i being the answer given candidate j and the cloze. We opt for multi-layer perceptrons (MLP) for implementation of g_θ.

4.3 Inferring

The purpose of this step is to piece together all the resulting clues obtained by comparing and to perform reasoning on them to make a choice. For the i-th candidate, we add up all its comparing results with others and pass the result through an inferring neural network to produce its final score:

$$s_i = f_\phi(\sum_{j=1}^{k} \mathcal{S}_{ij}), \tag{9}$$

where f_ϕ is a linear layer. This inferring process is executed for each candidate separately, each obtaining a likelihood score. For convenience of training, the final scores are passed through a softmax layer to generate a distribution $\hat{s}$ over the candidates. The candidate with the highest probability is chosen as the answer. Accordingly, the objective of training is to maximize the following function:

$$L = \sum_i log(\hat{s}_{a_i}), \tag{10}$$

where a_i is the correct answer.

5 Additional Features

In addition to the features introduced above, there are also some other useful features in previous works. Among them, two are incorporated into our model. Firstly, like Dhingra et al. (2017), we also find the *question evidence common word feature (qe-comm)* helpful to further boost the performance of our model. A *qe-comm* is defined by a one-hot vector $f_i \in \{0, 1\}^2$ for each document token w_i, indicating whether it is also present in the query. The *qe-comm* embedding $\widetilde{qe}_i$ for w_i can be obtained by $\widetilde{qe}_i = f_i^T F$, where $F \in R^{2 \times 2}$ is a feature lookup table and is then appended to the inputs $G \in R^{8d \times m}$ of the bidirectional matching layer for each document token. Specifically, we obtain $\widetilde{g_i}$ by concatenating g_i and $\widetilde{qe}_i$ for token w_i. Accordingly, the original inputs $G \in R^{8d \times m}$ are converted into $\widetilde{G} \in R^{(8d+2) \times m}$.

Secondly, character-level word embeddings have been widely used to alleviate the problem of modeling out-of-vocabulary (OOV) tokens (Yang et al., 2016). Let $w = [a_1, a_2, ..., a_l]$ be a word in the document or query, where a_i is a character of the word. To compute the character-level word embedding of token w, we first map each character a_i into a continous representation by means of a similar process as word embedding, then pass these continous representations through a Bi-GRU, and finally take the final output in forward direction as character-level word embedding.

6 Experiments and Results

In this section, we first introduce the datasets used for extensive experiments and then report the results.

6.1 Datasets

We evaluate the McR^2 model on four datasets: Children's Book Test (Named Entity), Children's Book Test (Common Noun), Who Did What (Strict), and Who Did What (Relaxed). The former two are developed from two subsets of the Children's Book Test (CBT) (Hill et al., 2016). A document in CBT is comprised of 20 contiguous sentences from the body of a popular children book, and a query is formed by displacing a token in the 21^{st} sentence with a placeholder. Following Hill et al. (2016), our experiments are only conducted on the subsets whose replaced token is either a named entity (NE) or a common noun (CN), because even a simple language model can already achieve high performance on the other types.

The other two datasets are constructed from Who Did What (WDW) (Onishi et al., 2016). Each WDW sample consists of two independent articles, one given as context document and the other on the same events as query. Deleted tokens in this corpus are always person named entities. In addition, samples that can be easily solved by simple baselines have been filtered, making the task more challenging. There are two versions of training set, WDW-Strict and WDW-Relaxed, in company with the same development and test sets. WDW-Strict is a small but tidy training set while WDW-Relaxed is larger and noisier. Our model is trained on both for respective results on the same validation and test sets.

6.2 Experimental Setups

We initialize all word embeddings with pre-trained GloVe vectors (Pennington et al., 2014). All hidden states of Bi-GRUs have 128 dimensions except those of 75 dimensions for character Bi-GRU. The internal weights of GRUs are initialized with random orthogonal matrices and the gradient clipping threshold is set to 5 in order to deal with gradient exploding issues with GRU units. We adopt the ADAM optimizer (Kingma and Ba, 2014) for weight updating with an initial learning rate of 0.001 and apply dropout (Srivastava et al., 2014) to word embeddings to avoid overfitting. We present two variants of our model, one using 100-dimensional GloVe vectors to initialize the word embeddings that are concatenated with character embeddings and the other with 300-dimensional GloVe vectors alone.

6.3 Overall Results

We compare the performance of our model and several state-of-the-art models. The results of them are collected from (Dhingra et al., 2017; Munkhdalai and Yu, 2016; Ghaeini et al., 2018). Table 1 presents the validation and the test accuracy of all these models on the datasets of CBT-NE, CBT-CN, WDW-Strict, and WDW-Relaxed. The comparison shows that our McR^2 (300*d*) outperforms all single models on the four datasets, giving a boost of 3.49 and 3.03 percentage points in validation/test accuracy on the WDW-Strict dataset over DGR, the best model so far. Interestingly, even the lighter version of our McR^2 (100*d*+*char_emb*) also gives state-of-the-art results on all the datasets except the development set of CBT-CN. Despite a single model, McR^2 shows superior performance over the best ensemble models, successfully demonstrating its advantages and effectiveness.

6.4 Ablation Study

We conduct a comprehensive ablation study to examine the effect of several key components of our model. Note that DGR (Ghaeini et al., 2018) is one of the best models for this task. To validate our multichoice relational reasoning module, we replace it with the pointer-sum attention mechanism suggested by (Kadlec et al., 2016). As shown in Table 2, the new model, McR^2 ($-mr$), results in a substantial performance drop on the CBT datasets. This proves the validity of performing reasoning on candidates. To

Model	CBT-NE		CBT-CN		WDW-Strict		WDW-Relaxed	
	dev	test	dev	test	dev	test	dev	test
Humans (Hill et al., 2016)	–	81.6	–	81.6	–	84.0	–	–
AS Reader (Kadlec et al., 2016)	73.8	68.6	68.8	63.4	–	57.0	–	59.0
IAA Reader (Sordoni et al., 2016)	75.2	68.6	72.1	69.2	–	–	–	–
EpiReader (Trischler et al., 2016)	75.3	69.7	71.5	67.4	–	–	–	–
AOA Reader (Cui et al., 2017)	77.8	72.0	72.2	69.4	–	–	–	–
GA Reader (Dhingra et al., 2017)	78.5	74.9	74.4	70.7	71.61	71.2	72.16	72.6
Fine-grained Gate (Yang et al., 2016)	79.1	74.9	75.3	72.0	–	71.7	–	72.6
DGR (Ghaeini et al., 2018)	77.9	75.4	73.8	70.7	71.78	72.0	72.26	72.9
AS Reader* (Kadlec et al., 2016)	76.2	71.0	71.1	68.9	–	–	–	–
EpiReader* (Trischler et al., 2016)	76.6	71.8	73.6	70.6	–	–	–	–
IAA Reader* (Sordoni et al., 2016)	76.9	72.0	74.1	71.0	–	–	–	–
AOA Reader* (Cui et al., 2017)	78.9	74.5	74.7	70.8	–	–	–	–
NSE (T=1) (Munkhdalai and Yu, 2016)	76.2	71.1	72.8	69.7	65.1	65.5	66.4	65.3
NSE Query Gating (T=12)	77.7	72.2	74.3	71.9	65.2	65.5	65.7	65.4
NSE Adaptive Computation (T=12)	78.2	73.2	74.2	71.4	66.5	66.2	67.0	66.7
AttSum-Feat (Hoang et al., 2018)	77.8	72.36	–	–	–	–	–	–
AttSum-Feat + L^1 (Hoang et al., 2018)	78.40	74.36	–	–	–	–	–	–
AttSum-Feat + L^2 (Hoang et al., 2018)	79.40	72.40	–	–	–	–	–	–
McR^2 (100*d*+*char_emb*)	**79.90**	75.96	74.50	72.52	73.60	73.15	72.74	73.24
McR^2 (300*d*)	79.65	**76.28**	**75.55**	**73.04**	**75.27**	**75.03**	**73.80**	**74.06**

Table 1: Validation/test accuracy (%) on CBT and WDW, with overall best results in bold. Note that 300*d* and 100*d* denote 300-dimentional GloVe vectors and 100-dimentional GloVe vectors, respectively, and *char_emb* denotes character embeddings. Model with (*) is ensemble model.

test the hierarchical multi-stage encoding architecture of McR^2, we replace it with the multi-hop representation mechanism used in DGR. The results in Table 2 show that McR^2 $(-fus)$ achieves an inferior performance to that of regular McR^2, indicating the usefulness of our encoding architecture.

As mentioned above, our model includes some other features like *qe-comm* and pre-trained GloVe vectors. Here we conduct an ablation test to analyze their contribution. As shown in Table 2, the results show a substantial performance drop for updating pre-trained GloVe vectors during training. This outcome agrees with the view that the prior knowledge provided by GloVe vectors is accountable for this performance drop (Dhingra et al., 2017). Furthermore, we can see from the last row that *qe-comm* also plays a significant role to boost the model's performance, a result consistent with previous works.

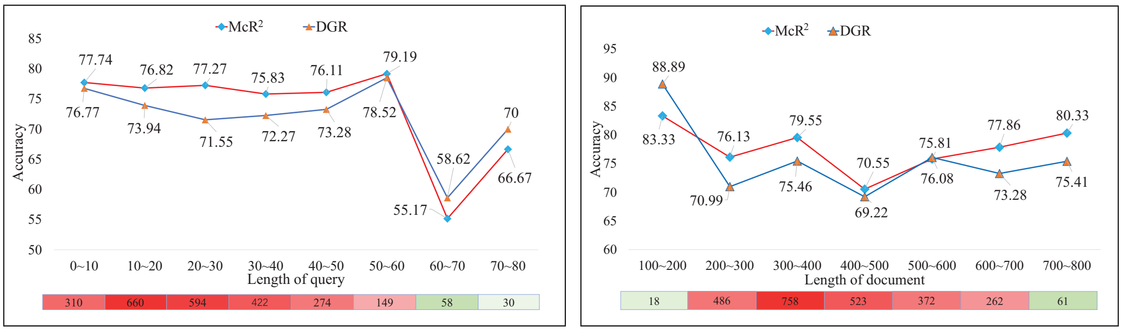
Model	CBT-CN		CBT-NE	
	dev	test	dev	test
McR^2	**75.55**	**73.04**	**79.65**	**76.28**
McR^2 (*−mr*)	74.45	72.0	78.55	74.24
McR^2 (*−fus*)	74.65	72.32	78.40	75.68
McR^2 (*update*)	73.65	70.28	78.65	74.36
McR^2 (*−qe-comm*)	75.15	72.16	79.05	75.48

Table 2: Results of ablation study.

6.5 Performance *vs* Text Length

In this subsection, we examine the relation between input length and model performance on the CBT-NE test set, using the strong DGR model for comparison. The results of comparison are presented in Figure 2a and 2b, from which we can see the McR^2 model's highly competitive performance against DGR at all lengths of document and query in the test set. In particular, when query length is under 60, which counts for about 90% of all queries, McR^2 exhibits a consistent outperformance over DGR. A reasonable explanation for the underperformance at query lengths beyond 60 is that condensing a too long query into a single vector brings noise to its candidate interaction module.

Furthermore, the performance comparison in Figure 2b shows that McR^2 does not surpass DGR only at two of the seven intervals of document length, including the shortest, revealing that our model is



(a) Length of query

(b) Length of document

Figure 2: Comparison of accuracy along the length of query and document, where the bar below the figure denotes the number of samples in a length interval of 10 and 100 words.

particularly good at handling long documents. The reason why DGR performs less well is most likely due to the incapacity of its basic architecture, the RNN, to learn long-range dependencies in long sequences. As a result, the representations of long documents produced by RNN are hence inevitably defective and affect the pointer-sum attention mechanism accordingly. In contrast, our candidate interaction module is able to provide extra clues through interaction between candidates, making the McR^2 relatively more resistant to this kind of deficiency.

6.6 Case Study

We further give an example to illustrate the effect of our reasoning module. For this purpose, we replace the multi-choice relational reasoning module with the pointer-sum attention (Kadlec et al., 2016), a mechanism conventionally used for this task. As shown in Figure 3, while the McR^2 model manages to choose the right answer, its variant fails to do that. After analysis, we tend to attribute the failure to the over attention of the variant to its fitness for candidate "George W. Bush" and the query and document, as clued by the highlighted span, since this candidate matches the two better. In contrast, our reasoning module is able to perform comparing among candidates based on query and document evidences and deduce the correct answer finally.

Since Equation 9 includes a linear function f_ϕ, we can expand it as follows: $s_i = f_\phi(\sum_{j=1}^{k} \mathcal{S}_{ij}) = \sum_{j=1}^{k} f_\phi \mathcal{S}_{ij} = \sum_{j=1}^{k} s_{ij}$, where s_{ij} can be considered the "likelihood" of candidate i being an answer given candidate j, or a "supporting score" from candidate j. To be more vivid, we plot the supporting scores of each candidate of the above example in Figure 4, from which we can note that candidate B clearly receives the largest overall score among all the candidate answers. This explicitly explains why it is chosen as the final answer by our model.

Document: [...] Putin urged President-elect Barack Obama to drop the plan US missile shield in Eastern Europe. [...] Obama has yet to give firm details over whether he intends to continue plan created by administration of Republican President George W. Bush. Russian President Dmitry Medvedev said at the weekend that while the Bush administration's position looked "extremely inflexible" then "the position of the President-elect looks more careful."

Query: Russian President Dmitry Medvedev said Sunday he believed ___ would be open to changing position over a hotly contested US plan for a missile defense shield in Eastern Europe.

Candidates: Vladimir Putin | Barack Obama | George W. Bush;

Reference: Barack Obama | **Variant**: George W. Bush | **McR^2**: Barack Obama

Figure 3: An example for case study. The variant is obtained by replacing the multi-choice relational reasoning module of McR^2 with the pointer-sum attention (Kadlec et al., 2016).

6.7 Time Analysis

Usually a neural network model spends most of its training time on the representation phase. Although involving a hierarchical multi-stage architecture, our fusion representation module may consume less

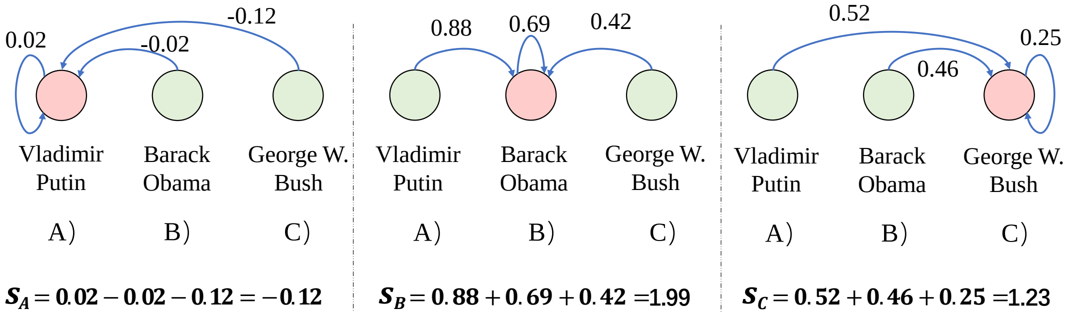


Figure 4: Supporting scores of each candidate answer from others.

training time than those more complicated representation mechanisms such as the one of DGR. To exemplify this, we build a variant of McR^2 whose hierarchical multi-stage architecture is replaced with the multi-hop representation of DGR, and compare its training time against the regular McR^2 on each batch of size 32. The result shows that the regular McR^2 takes only 24 seconds on average while its variant takes 45 seconds, indicating that the latter costs 87.5% more training time than McR^2.

6.8 Pre-trained Language Models Encoder

Huge pre-trained language models like BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) have produced promising results in many natural language processing tasks. They are trained on a mass of general-domain data and then fine-tuned on downstream problems. For this cloze-style reading comprehension taks, the performance is also outstanding. For example, GPT-2 (Radford et al., 2019) obtains new-state-of-the-art results of 87.65% on the development dataset of CBT-CN and 83.4% on the development dataset of CBT-NE, as reported in the paper, which is even better than human performance.

Here we study the effect of applying these powerful language models in our framework. We replace our lightweight fusion representation module with the BERT encoder. In particular, we first utilize BERT to encode document and query respectively, and obtain their contextual representations. Then, these contextual representations are taken as input to our multi-choice relational reasoning module to deduce the correct answers. As BERT can only encode a piece of text no more than 512 words at a time, and documents in this task are usually longer than that, we first cut each document into several segments and input them into BERT in turn. After that, we concatenate the outputs together as the final contextual representation for the document. Note that it does not seem rational to simply discard the part of document that exceeds the maximum length because candidates may occur in any positions of document.

Model	Accuracy (%)	
	CBT-CN	CBT-NE
McR^2	74.5	**79.9**
Variant	**79.3 (+4.8)**	78.1 (-1.8)

Table 3: Results of a variant of McR^2 on the CBT datasets.

As for a query, we directly utilize BERT for the encoding as it is usually very short in length. We evaluate this variant on the development datasets of CBT. As shown in Table 3, the variant of our model obtains a result of 79.3 on CBT-CN and 78.1 on CBT-NE. Obviously, using this powerful model to implement our encoding module is able to boost the performance of this task.However, the performance on the CBT-NE dataset is inferior to that of our McR^2. The possible reason behind is that the variant lacks interaction for document and query when utilizing BERT for encoding, while the attention mechanism in our fusion representation module is able to make up the defect to a certain extent. The above experiments prove that our approach is able to improve on top of frontier models as well.

7 Related Work

Existing approaches to cloze-style reading comprehension can be categorized into the following categories in terms of the methodologies used to generate answers.

Multi-classification. Hermann et al. (2015) introduced three multi-classification models, including DeepLSTM, Attentive Reader, and Impatient Reader. A common pipeline for these models is first to produce contextual representations via long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and attention mechanism (Bahdanau et al., 2014) and then compute a joint document-query representation and pass it through a linear layer to predict the probability that a word in the vocabulary is a true answer. In essence, reading comprehension is regarded as a multiple classification problem.

Attention Sum. Kadlec et al. (2016) presented a simple model Attention-Sum (AS) Reader that initially uses two bidirectional GRUs (Bi-GRU) to encode query and document independently. Then, it computes a probability distribution over all document tokens by the softmax of the dot product between query and token representations. Finally, AS Reader aggregates the probabilities of the same candidate in multiple appearances with the aid of a pointer-sum attention mechanism which has been adopted by many subsequent models (Sordoni et al., 2016; Cui et al., 2017; Dhingra et al., 2017; Ghaeini et al., 2018). In addition, these models also explore different ways of fusion representations for document and query. For example, Sordoni et al. (2016) introduced an iterative alternating attention mechanism Iterative Alternative Attention (IAA) Reader that allows a fine-grained exploration of both query and document rather than condensing a query into a single vector. Cui et al. (2017) proposed a two-way attention mechanism, the Attention-over-Attention (AoA) Reader, to allow query and document to mutually attend to each other, which appears to be effective in model training. Dhingra et al. (2017) introduced Gated-Attention (GA) Reader with a gated-attention mechanism implemented in two steps. First, the attentive interaction between each intermediate state of a RNN document encoder and all query words is modeled to yield an attended query representation. Then, a multiplicative operation is executed on the attended query representation and the RNN intermediate state. A deficiency of GA Reader is that the query encoding is independent of its previous iterations. To alleviate this, Ghaeini et al. (2018) extended GA Reader to encode a query depending not only on document but also on the reading of previous iterations.

Hypothesis Testing. Hypothesis testing is an important way of reasoning for human beings. Motivated by this, Sordoni et al. (2016) proposed EpiReader that first applies AS Reader to generate a small set of candidates, and then formulates a few hypotheses, each by replacing a query placeholder with a selected candidate. EpiReader reranks hypotheses in terms of their entailments about a document and designates the candidate with the highest entailment score as the final answer. Based on memory augmented neural networks, Munkhdalai and Yu (2016) (NSE) proposed a hypothesis testing framework that gradually refines previously formed hypotheses into new one for testing.

Other Methods. In their work Hoang et al. (2018) focused on hard entity tracking cases with additional entity features and trained their model with a multi-task tracking objective. Their approach is shown the ability to enhance long-term dependencies of words and improve task performance. A recent trend for this task trends to rely on pre-trained language models. For example, Radford et al. (2019) proposed the GPT-2 language model that is pre-trained on a mass of general-domain data. GPT-2 has demonstrated very impressive performance in a wide range of tasks including reading comprehension.

8 Conclusion

In this paper, we have proposed and tested a novel multi-choice relational reasoning (McR^2) model for cloze-style reading comprehension. McR^2 is built upon effective hierarchical multi-stage representations of document and query, a novel pointer-sum vector layer to aggregate representations of candidate answers, and relational reasoning on candidate answers to piece together evidence from all candidates. Our experiments on several benchmark datasets show that this model compares favorably against state-of-the-art models. An ablation study also confirms the effectiveness of its key components and a time analysis further provides evidence for its efficiency. In essence, McR^2 adopts an extract-then-reason framework. Although introduced for cloze-style reading comprehension task in this paper, it can be applied to many other tasks for which interaction among multiple candidates is needed to deal with a single query.

Acknowledgments

We thank the anonymous reviewers for their constructive reviews. This work was supported by the Fundamental Research Funds for the Central Universities (No.19lgpy220) and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602, Vancouver, Canada, July.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada, July.

Reza Ghaeini, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2018. Dependent gated reading for cloze-style question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3330–3345, Santa Fe, New Mexico, USA, August.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *ICLR*.

Luong Hoang, Sam Wiseman, and Alexander Rush. 2018. Entity tracking improves cloze-style reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1055, Brussels, Belgium, October-November.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918, Berlin, Germany, August.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tsendsuren Munkhdalai and Hong Yu. 2016. Reasoning with memory augmented neural networks for language comprehension. *CoRR*, abs/1610.06454.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas, November.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October.

Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. 2017. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Adam Trischler, Zheng Ye, Xingdi Yuan, Philip Bachman, Alessandro Sordoni, and Kaheer Suleman. 2016. Natural language comprehension with the EpiReader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Austin, Texas, November.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada, July.

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W Cohen, and Ruslan Salakhutdinov. 2016. Words or characters? fine-grained gating for reading comprehension. *arXiv preprint arXiv:1611.01724*.