Towards Human-level Machine Reading Comprehension: Reasoning and Inference with Multiple Strategies

Yichong Xu¹, Jingjing Liu², Jianfeng Gao², Yelong Shen², and Xiaodong Liu²
¹ Carnegie Mellon University
² Microsoft AI&R
yichongx@cs.cmu.edu
{jingjl, jfgao, yeshen, xiaodl}@microsoft.com

Abstract

This paper presents a new MRC model that is capable of three key comprehension skills: 1) handling rich variations in question types; 2) understanding potential answer choices; and 3) drawing inference through multiple sentences. The model is based on the proposed MUlti-Strategy Inference for Comprehension (MUSIC) architecture, which is able to dynamically apply different attention strategies to different types of questions on the fly. By incorporating a multi-step inference engine analogous to ReasoNet (Shen et al., 2017), MUSIC can also effectively perform multi-sentence inference in generating answers. Evaluation on the RACE dataset shows that the proposed method significantly outperforms previous state-of-the-art models by 7.5% in relative accuracy.

1 Introduction

Machine Reading Comprehension, along with the traditional QA task, has been recently revolutionized by the deep learning wave (Dhingra et al., 2016; Seo et al., 2016; Shen et al., 2017). Specifically, state-of-the-art MRC systems have reached near-human performance on SQuAD dataset (Rajpurkar et al., 2016). However, SQuAD alone is not enough for complete machine comprehension: 74.1% of the questions in SQuAD can be trivially solved via word-based search and context-matching without deeper reasoning (Trischler et al., 2016). A recent analysis shows that SQuAD and several other popular datasets require limited comprehension skills from MRC systems (Sugawara et al., 2017), as manifested in their suffered performance when applied to more realistic MRC tasks, such as TriviaQA (Joshi et al., 2017) from Trivia competitions and RACE (Lai et al., 2017) from exam questions. For example, the exact-match score of M-Reader ((Hu et al., 2017), one of the best models on SQuAD) drops from 77.7 to 54.5 when applied to TriviaQA, whereas human performance is 82.3 and 79.7 for SQuAD and TriviaQA, respectively.

This paper strives to take a step towards developing an intelligent system capable of comprehending text as human. To this end, we focus on questions from human English examinations, i.e., the RACE dataset (Lai et al., 2017), because they are specifically designed to test students’ comprehension ability in various aspects, which provides more comprehensive and realistic evaluation on MRC than other popular datasets such as SQuAD.

We find that exam questions have broader topic coverage and require deeper reasoning than questions from existing MRC datasets. For example, Figure 1 compares questions from English exams in Chinese high schools and those from SQuAD.

The story is mainly about ___.
What was the author’s attitude towards the industry awards for quieter?
The following is true EXCEPT ___.
The idea of using stamps was thought of by ___.

What was the final score of Super Bowl 50?
In which point did the drainage basin of the Amazon split?
Where did the water in the Amazon Basin flow towards when moving west?

Figure 1: Above: Questions from English exams. Below: Questions from SQuAD.
These English exam questions require more sophisticated reading comprehension skills such as summarization (in the first question), inference (in the second question) and deduction (in the third question).

In this paper, we present a new neural MRC model that is capable of: (1) handling rich variations in question types; 2) understanding answer choices; and 3) drawing inference through multiple sentences. Our model is based on a new MRC architecture called Multi-Strategy Inference for Comprehension (MUSIC). Different from other MRC models that use a fixed attention strategy for each question, MUSIC can dynamically apply different attention strategies to different questions on the fly: each strategy can apply a unique attention model to passages, questions and answer choices. A matching gate trained by reinforcement learning automatically controls which strategy to apply in each iteration. We also use a multi-step inference engine, with a termination gate trained by reinforcement learning, to dynamically control how many inference steps the system should take before reaching the right answer. We evaluate our model on RACE (Lai et al., 2017), a dataset consisting of questions from middle and high school English exams in China. Experiments show that the proposed MUSIC approach outperforms previous state-of-the-art models by 7.5% in relative accuracy.

The rest of this paper is organized as follows. We revisit related work in MRC in Section 2. In Section 3, we identify the key challenges in solving real reading comprehension questions in human tests; we then present our proposed model with multi-strategy inference in Section 4. Experimental results with detailed analysis are provided in Section 5.

2 Related Work

The recent progress in MRC is largely due to the introduction of large-scale datasets. CNN/Daily Mail (Hermann et al., 2015) and SQuAD (Rajpurkar et al., 2016) are two popular datasets that are widely used. More recently, other datasets have been introduced, such as MS MARCO (Nguyen et al., 2016), NewsQA (Trischler et al., 2016) and RACE (Lai et al., 2017). MS MARCO collects data from search engine queries and user-clicked results, thus contains a broader topic coverage than Wikipedia and news articles in SQuAD and CNN/Daily Mail. NewsQA separates the data annotation process into question collection and question answering, which makes the generated questions more natural. RACE focuses primarily on comprehension skills, and uses questions from English language exams that are designed to test human comprehension skills. In this paper we focus on RACE, as our goal is to develop new MRC systems for human-level comprehension tasks.

As summarized in (Weissenborn et al., 2017), a typical MRC system consists of three components: firstly, an encoder integrates raw words and characters into a concatenated representation, typically using word embeddings and recurrent neural networks (RNNs), e.g., LSTM. Secondly, a matcher applies attention mechanism between the question and the passage. Thirdly, an answerer uses the attended representation to select the right answer from a pool of potential answers. Most previous work used this generic framework. For example, BiDAF (Seo et al., 2016) uses Glove (Pennington et al., 2014) embeddings and character encodings with LSTM as the encoder. Then the passage and questions are matched and combined in a bi-directional fashion. Finally, two softmax functions over all locations in the passage act as the answerer for selecting spans. ReasoNet (Shen et al., 2017) employs an encoder and a matcher similarly, but uses a multi-step inference answerer to select spans.

The main body of research on MRC focuses on the attention mechanism. Attention involves computing a similarity score between the question and passage representations, and then updating these representations using the similarity scores. For example, Chen et al. (2016) proposes computing a bilinear form of similarity scores and using these scores to compute a weighted sum of passage representations. BiDAF (Seo et al., 2016) uses a bi-directional attention, where the weighted sum of both passage and question representations are computed. BiMPM (Wang et al., 2017) proposes a multi-perspective matching, where each perspective produces a similarity matrix using different linear transformations of question/passage representations. Our multi-strategy attention is not confined by the specific implementation of a single attention; instead, our model is generic in the sense that it can use any two-component matcher as building blocks.
3 Key Challenges in Solving Human Comprehension Tests

In this section, we identify three key skills of an MRC system to solve human comprehension tests. We use RACE as our reference, which is a recently released MRC dataset consisting of 97,867 questions from English exams in Chinese middle and high schools. It was divided into two subsets, RACE-M and RACE-H, from middle school and high school exams. RACE-M has 28,293 questions and RACE-H has 69,574.

**Variety in question types.** As stated in Section 1, RACE contains a much broader spectrum of question types than other MRC datasets. Different comprehension skills are required for handling different questions, such as skimming, summarizing, inference and deduction. It is thus important for MRC systems to be able to adapt for different question types.

**Complexity of answers.** Figure 2 shows some examples of RACE and CNN questions. The answer choices in RACE are natural language sentences, whereas answers in CNN are entities. In other datasets (e.g., SQuAD), answers are often spans in the passage. The natural language form of answer choices in RACE lead to yet another difficulty: besides understanding questions and passages, the MRC system also needs to understand the answer choices.

**Multi-step inference.** Inference refers to the skill of making connection between sentences in a passage, and inferring the answer by summarizing information throughout the passage. Table 1 shows a comparison among different datasets on the level of requirement for inference. The low numbers in SQuAD and CNN show that inference skills are understated in these datasets, whereas answers are often natural language sentences, the MRC system also needs to understand the answer choices.

Table 1: Percentage of questions in each dataset that requires Single-sentence Inference (SI) and Multi-sentence Inference (MI), from (Lai et al., 2017).

| Dataset | % SI questions | % MI questions |
|---------|----------------|----------------|
| CNN     | 19.0%          | 2.0%           |
| SQuAD   | 8.6%           | 11.9%          |
| NewsQA  | 13.2%          | 20.7%          |
| RACE-M  | 31.3%          | 22.6%          |
| RACE-H  | 34.1%          | 26.9%          |
| RACE    | 33.4%          | 25.8%          |

4 Multi-Strategy Inference for Comprehension (MUSIC)

In this section, we present a new model for MRC: Multi-Strategy Inference for Comprehension (MUSIC). The overall architecture of MUSIC is depicted in Figure 3. Input is a question \( Q \) of length \( l_q \), a passage \( P \) of length \( l_p \), and a list of \( r \) potential answers \( \mathcal{A} = \{A^1, ..., A^r\} \) of length \( l_a^1, ..., l_a^r \). The architecture consists of a standard Embedding layer and a Context layer, on top of which is a Gated Matching layer to apply different matching strategies to different questions. A multi-step inference layer is also incorporated for multi-step reasoning. The output of the model is an answer choice \( C \in \{1, 2, ..., r\} \). Details of the architecture and the training method are presented in the following sub-sections.

4.1 Model Architecture

**Embedding Layer and Context Layer.** Different from previous models, we transform each answer choice in \( \mathcal{A} \) along with the question \( Q \) and the passage \( P \) in the embedding and context layers. We first transform each word in \( A, Q, P \) into its 300-dimension GloVe embedding (Pennington et al., 2014). Following previous literature (Wang et al., 2017), we also embed each character into a 20-dimension character embedding. After concatenating the character embedding and word embedding, we obtain embeddings of words in \( P, Q, A \) and \( \mathcal{A} \). Let \( Q = \{q_{ij}\}_{i=1}^{l_q}, P = \{p_{ij}\}_{i=1}^{l_p} \), and \( A_j = \{a_{i,j}\}_{i=1}^{l_a} \) for each \( j = 1, 2, ..., r \).

Then in the context layer, we pass all these representations into a bi-directional LSTM to obtain context representations. Since answers are not always complete sentences, we append the question before each answer and feed the concatenated sentence into LSTM. The obtained context vectors are represented as

\[
Q^c = \text{LSTM}(Q) = \{q_{ij}^c, q_{ij}^t\}_{i=1}^{l_q},
\]

\[
P^c = \text{LSTM}(P) = \{p_{ij}^c, p_{ij}^t\}_{i=1}^{l_p},
\]

\[
(Q + A)^j = \text{LSTM}(Q + A_j) = \{c_{ij}^e, c_{ij}^t\}_{i=1}^{l_q + l_a},
\]

\[
j = 1, 2, ..., r.
\]

**Gated Matching Layer.** This is the core of
Raising pets is a popular online game among teenagers. ... You can feed, wash, talk to and play with your pet ...

Question: What does the passage mainly talk about?

Choices:
A. Raising pets online is popular among teenagers.
B. It’s bad to raise pets online.
C. How to raise pets online.
D. It’s good to adopt pets online.

Passage: ...@entity0 called me personally to let me know that he would not be playing here at @entity23 , ” @entity3 said on his @entity21 event ’s website...

Question: @placeholder says he understands why @entity0 won’t play at his tournament

Choices: @entity0, @entity3, @entity23, @entity21,....
Figure 3: Architecture of MUSIC. i) Passage, question and answer choices are mapped through word and character embeddings in the Embedding Layer. ii) The embeddings are then fed into a Bi-LSTM in the Context Layer. iii) The Gated Matching Layer makes customized attention matchings across the three representations from passage, question and answer choices. iv) The Multi-step Reasoning Unit reads through the memory for a dynamic number of steps. v) Answer prediction gives the final answer.

rate terminate gates for each answer, to contain the size of the action space and the variance in training. Since answers are mutable, the input weights for each answer fed into the gate softmax are the same.

\[ p_t, 1 - p_t = \text{softmax}(\sum_{j=1}^{T} W_2 s^j_t), t = 1, 2, ..., T, \]
\[ T_t = \text{Bernoulli}(p_t), \]
\[ T = \min\{t : T_t = 1\}. \]

Answer Prediction. Finally, an answer prediction is obtained by applying a softmax over the final state for each answer:

\[ C = \text{Category}(\text{softmax}(W_3 s^T), \text{softmax}(W_3 s^T_2, ..., \text{softmax}(W_3 s^T_r))), \]

where \( W_3 \) transforms \( s^T_j \) into a numerical logit for each answer. Again, since the answers are mutable, \( W_3 \) is the same across all answers.

4.2 Training details

Since the gate choice and termination steps are random variables, MUSIC cannot be directly optimized by backpropagation. Instead, we see \( G \) and \( T \) as policies, and use the REINFORCE algorithm (Williams, 1992) to train the network by policy gradient. We define the reward \( r \) to be 1 if \( C \) (final answer) is correct, and 0 otherwise. Each possible value pair of \((C, G, T)\) corresponds to a possible episode, which leads to \( r \cdot n \cdot T \) possible episodes. Let \( \pi(c, g, t; \theta) \) be any policy parameterized by MUSIC parameter \( \theta \), and \( J(\theta) = E_\pi[r] \) be the expected reward. Thus

\[ \nabla_{\theta} J(\theta) = E_{\pi(g,c,t;\theta)} \left[ \nabla_{\theta} \log \pi(c, g, t; \theta)(r - b) \right] \]
\[ = \sum_{g,c,t} \pi(g, c, t; \theta) \]
\[ \nabla_{\theta} \log \pi(c, g, t; \theta)(r - b), \quad (1) \]

where \( b \) is some critic value function. Following (Shen et al., 2017), we set \( b = \sum_{g,c,t} \pi(g, c, t; \theta)r \) and replace the \((r - b)\) term above by \((r/b - 1)\) to achieve better performance.

5 Experiments

In this section, we present the experimental results along with implementation details.
5.1 Implementation Details

Matching Strategies. We choose matching strategies inspired by the BiMPM model (Wang et al., 2017), although our model can accommodate any matching strategies. Each matching strategy (specified below) produces two attention results \(Q^m, A_j^m\), which is of length \(l_q\) and \(l_t\), respectively; then two bi-directional LSTMs (their weights are not tied) are applied to \(Q^m\) and \(A_j^m\).

\[
\begin{align*}
Q^m, A_j^m &= f_k(Q^c, P^c, (Q + A)^c_j), \\
Q^a &= \text{LSTM}_1(Q^m), \\
A_j^a &= \text{LSTM}_2(A_j^m).
\end{align*}
\]

The output is the last output state of \(Q^a\) and \(A_j^a\).

We now define \(f_k\) for each strategy. Wang et al. (2017) defines the multi-perspective matching operator \(\otimes\) that operates between any two sentences (texts) \(P\) and \(Q\). \(P \otimes Q\) produces matching results of matching \(P\) to \(Q\) and reversely, which are represented as \(P \bowtie Q\) and \(Q \bowtie P\), respectively. Briefly, \(P \bowtie Q\) computes a multi-perspective similarity function of each word representation in \(Q\) with respect to a summary (the last word representation, the max pooling, the attention sum, and the max attention sum) of all words in \(P\). The \(k-th\) dimension (perspective) of the similarity function is computed by \(\cos(W_k \circ v_p, W_k \circ v_q)\), where \(v_p, v_q \in \mathbb{R}^d\) are the representations of words in \(P\) and \(Q\), and \(W_k \in \mathbb{R}^d\) is a \(d\)-dimensional trainable parameter. \(P \bowtie Q\) has the same length \(l_q\) as \(Q\), and represents the similarity between each word in \(Q\) and the whole sentence \(P\). The actual form of matching is not crucial to our system, so we refer readers to (Wang et al., 2017) for details of \(\otimes\). We use the number of matching strategies \(n = 3\) in our implementation, and define the matching functions as

\[
\begin{align*}
Q_j^c, A_j^c &= \text{Split} \left( ((Q + A)^c_j) \right), \\
f_1 &= \text{Split} \left( P^c \bowtie (Q + A)^c_j \right), \\
f_2 &= Q^c, P^c \bowtie A_j^c, \\
f_3 &= (P^c \bowtie A_j^c) \bowtie (P^c \bowtie Q)^j, \\
(P^c \bowtie A_j^c) \bowtie (P^c \bowtie A_j^c).
\end{align*}
\]

Here the function \(\text{Split}(\cdot, \cdot)\) takes a representation of length \(l_q + l_t\) and splits it into two representations of length \(l_q\) and \(l_t\). The specific choice of matching gates are made to fit different questions: Matching 1 aims to fit short answer questions, where the answer \(A_j^c\) alone only makes sense when concatenated with the question. Matching 2 aims to fit longer answers, so that we can match \(A\) directly with \(P\). The question is left unchanged in Matching 2. Matching 3 is a more complex and deeper option. We first match the passage to the question and the answer choices; and then we apply a bi-directional matching between the two derived results. This matching is designed for those questions that require more inference or summarization.

Memory. The memory is computed similarly as the matching strategies using a matching function \(Q^c \bowtie P^c\), except that we take all the representation output of LSTM instead of using the last state:

\[
\text{Memory} = M^a = \text{LSTM}_3(Q^c \bowtie P^c).
\]

Multi-step Inference. We use the same attention function as in (Shen et al., 2017). An attention score \(a_{t,i}\) is computed based on memory \(m_i^a \in M^a\) and state \(s_t\) as \(a_{t,i} = \text{softmax}_{i=1,2,\ldots,l_m} \cos(W_4 m_i^a, W_5 s_t)\), where \(l_m = l_q\) is the memory length, and \(W_4, W_5\) are trainable weights. The attention vector is then computed as \(f_{\text{att}} = \sum_{i=1}^{l_m} a_{t,i} m_i^a\).

Parameter Setup. Most of our parameter settings follow (Wang et al., 2017) and (Shen et al., 2017). We use (1) to update the model, and use ADAM (Kingma and Ba, 2014) with learning rate 0.001 and batch size 64 for optimization. A small dropout rate of 0.1 is applied to each layer. All LSTMs have a hidden dimension of 100. The maximum reasoning step \(T\) is set to 5. We limit the length of passage/question/answer to a maximum of 500/100/100 for efficient computation. The model is implemented in Tensorflow (Abadi et al., 2016) and we plan to release the source code soon.

5.2 Model Performance

We compare the performance of MUSIC with baseline methods in Table 2. All models were trained with the whole RACE dataset, and tested on RACE-M and RACE-H, respectively. As shown in the table, on RACE-M, MUSIC leads to a 7.8% and 7.3% performance boost over GA and Stanford AR, and has a 1.5% and 2.7% outperformance on RACE-H.

Ablation Studies. For ablation studies, we trained 4 different models on RACE-M: 1) The full MUSIC model; 2) MUSIC without multi-strategy
In Nanjing... each student is tested on three sports. They can choose **long jump**, **basketball dribbling** or **volleyball**.

Q: The P. E. test in Nanjing includes all of these sports except ___.

A: 'skipping', 'basketball', 'football', 'volleyball'

**Steps:** 1

---

**P:** What are some successful foods? … Instant noodles have also become a global success.

Instant noodles were invented in Japan by Momofuku Ando. … University students like them a lot. However, instant noodles aren't innocent. They're high in fat and salt…

Q: The passage is mainly about ___.

A: 'how and where the instant noodles were invented', 'the advantages of the instant noodles', 'the disadvantages of the instant noodles', 'All of the above.'

**Steps:** 5

---

**P:** In many countries, holidays are important parts in people's life. American people... In Canada many people... People in France are very good at enjoying life. They have a 6-week holiday every year, and they work less than 40 hours a week.

Q: In which country can people work less than 40 hours a week?

A: 'America.', 'Canada.', 'France.', 'China.'

**Steps:** 1

---

**Figure 4:** Examples of choosing Matching Gate and number of inference steps in different questions types. Left: The question and the answer choice are part of a full sentence, a case handled by Gate 1, with one-step inference. Middle: A complex question with natural language answer choices, a case where Gate 2 was applied and 5 inference steps were required. Right: A question that requires deep inference, a case handled by Gate 3, with one-step inference.

---

**Table 2:** Accuracy% of MUSIC compared to baseline methods on RACE test sets. Results of baseline models come from (Lai et al., 2017) and unpublished (Anonymous, 2018). Note that ElimiNet<sub>Ensemble</sub> ensembles 6 equivalent models.

| Model          | RACE-M | RACE-H | RACE  |
|----------------|--------|--------|-------|
| Sliding Window | 37.3   | 30.4   | 32.2  |
| Stanford AR    | 44.2   | 43.0   | 43.3  |
| GA             | 43.7   | 44.2   | 44.1  |
| ElimiNet       | N/A    | N/A    | 44.5  |
| ElimiNet<sub>Ensemble</sub> | N/A    | N/A    | 46.5  |
| **MUSIC**      | 51.5   | 45.7   | 47.4  |

**Table 3:** Ablation studies of MUSIC for multi-strategy matching (MM) and multi-step reasoning (MR).

| Model                          | Accuracy% |
|--------------------------------|-----------|
| MUSIC                          | 51.3      |
| MUSIC without MM               | 49.4      |
| MUSIC without MR               | 50.4      |
| MUSIC without MM, MR           | 48.7      |

We trained 3 different models with each gate, and chose the best among them.

**Matching and Step choices.** Here we present a detailed analysis on how the system chooses different matching gates and various steps of inference for different questions. Figure 4 shows some randomly-chosen questions from RACE-M.

Recall that our three matching strategies are targeted for: 1) short answers; 2) long answers; and 3) deeper inference. We can see how the system made different choices on matching gates and inference steps for different types of questions. The first question has short answers, and both the question and answer choices are part of a full sentence. Thus, matching gate 1 was applied in this case. The question is relatively easy to answer, so single-step inference is sufficient. 2) The second question has long natural language answers, which can only be obtained from multiple sentences throughout the passage. This requires multi-step inference. So in this case, matching gate 2 was selected, and the system went through 5 steps of inference. The third question also requires inference (e.g., finding the resolution of "they" in the last sentence). While other matching choices (e.g., matching gate 1 with multi-step inference) are also possible, we suspect that the system chose gate 3 because the coreference resolution is a relatively easy inference and can be done with matching strategy 3.

6 Conclusion

In this work, we build a new MRC model that incorporates multiple paradigms of strategy for reasoning and inference for human-level reading comprehension. The proposed MUSIC architec-
ture adopts a multi-strategy mechanism by incorporating a gated-matching component and a multi-step reasoning component, to dynamically control what matching strategy and how many steps of inference to apply to each question on the fly. The matching gates and the termination gates for inference are trained jointly with reinforcement learning. Experiments on the RACE dataset show that our model can greatly outperform state-of-art approaches in real human comprehension tasks, with the capability of adapting for a variety of question types with dynamic inference. For future work, we will investigate how to make the architecture composable, and how to let the system learn new strategies from data automatically.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Anonymous. 2018. Eliminet: A model for eliminating options for reading comprehension with multiple choice questions. International Conference on Learning Representations https://openreview.net/forum?id=B1bgpzZAZ.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. arXiv preprint arXiv:1606.02858.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. arXiv preprint arXiv:1606.01549.

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. pages 1693–1701.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. CoRR, abs/1705.02798.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqua: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551.

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

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP). pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasont: Learning to stop reading in machine comprehension. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pages 1047–1055.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). volume 1, pages 806–817.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. arXiv preprint arXiv:1611.09830.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. arXiv preprint arXiv:1702.03814.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural qa as simple as possible but not simpler. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 271–280.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8(3-4):229–256.