applied sciences

MDPI

Review

Neural Machine Reading Comprehension: Methods and Trends

Shanshan Liu, Xin Zhang *, Sheng Zhang, Hui Wang and Weiming Zhang

Science and Technology on Information Systems Engineering Laboratory, College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; liushanshan17@nudt.edu.cn (S.L.); zhangsheng@nudt.edu.cn (S.Z.); huiwang@nudt.edu.cn (H.W.); wmzhang@nudt.edu.cn (W.Z.)

* Correspondence: ijunzhanggm@gmail.com

Received: 27 July 2019; Accepted: 3 September 2019; Published: 5 September 2019

check for updates

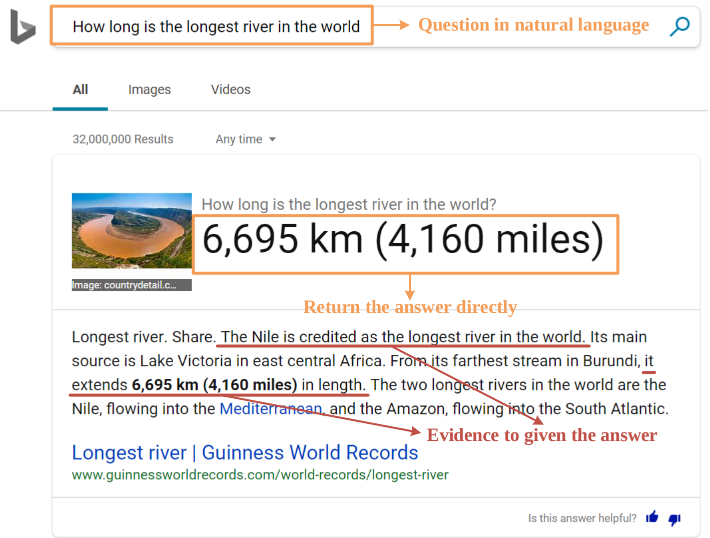
Abstract: Machine reading comprehension (MRC), which requires a machine to answer questions based on a given context, has attracted increasing attention with the incorporation of various deep-learning techniques over the past few years. Although research on MRC based on deep learning is flourishing, there remains a lack of a comprehensive survey summarizing existing approaches and recent trends, which motivated the work presented in this article. Specifically, we give a thorough review of this research field, covering different aspects including (1) typical MRC tasks: their definitions, differences, and representative datasets; (2) the general architecture of neural MRC: the main modules and prevalent approaches to each; and (3) new trends: some emerging areas in neural MRC as well as the corresponding challenges. Finally, considering what has been achieved so far, the survey also envisages what the future may hold by discussing the open issues left to be addressed.

Keywords: deep learning; machine reading comprehension; survey

1. Introduction

Machine reading comprehension (MRC) is a task introduced to test the degree to which a machine can understand natural languages by asking the machine to answer questions based on a given context, which has the potential to revolutionize the way in which humans and machines interact with each other. For example, as shown in Figure 1, a search engine with MRC techniques can directly return the correct answers to questions posed by users in natural language rather than a series of related web pages. In addition, smart assistants equipped with an MRC system can read help documents and provide users with high-quality consulting services. In sum, MRC is a promising task, which can make information retrieval more efficient.

Early MRC systems date back to the 1970s, the most notable of which was the QUALM system proposed by Lehnert [1]. However, restricted by its small scale and specific domain, this system could not be widely applied. Research on MRC was ignored in the 1980s and 1990s. In 1999, Hirschman et al. [2] released an MRC dataset, containing stories of third- to sixth-grade material with five "wh" (what, where, when, why, and who) questions, which again directed the research attention to MRC tasks. At that time, methods for solving MRC tasks were mainly rule-based or machine-learning-based: Hirschman et al. [2] proposed a bag-of-words technique that represented sentences with questions and context as sets of words and chose words occurring in both the question and context as the answer. Riloff et al. [3] designed a rule-based MRC system, Quarc, containing different heuristic rules with morphological analysis, such as part-of-speech tagging, semantic class tagging, and entity recognition, for different types of "wh" questions. Poon et al. [4] used a machine-learning method combining bootstrapping, Markov logic, and self-supervised learning for machine reading. However, those methods suffer from some, if not all, of the following limitations. First, they are mainly based on hand-crafted rules or features that require substantial human effort. Second, those systems are incapable of generalization, and their performance may degrade due to large-scale datasets of myriad types of articles. Finally, some traditional approaches not only ignore long-range dependencies but also fail to extract contextual information.


How long is the longest river in the world
Question in natural language
All
Images
Videos
32,000,000 Results
Any time
How long is the longest river in the world?
6,695 km (4,160 miles)
Image: countrydetail.c...
Return the answer directly
Evidence to given the answer
Longest river | Guinness World Records
www.guinnessworldrecords.com/world-records/longest-river
Is this answer helpful?

Longest river. Share. The Nile is credited as the longest river in the world. Its main source is Lake Victoria in east central Africa. From its farthest stream in Burundi, it extends **6,695 km (4,160 miles)** in length. The two longest rivers in the world are the Nile, flowing into the Mediterranean, and the Amazon, flowing into the South Atlantic.

Figure 1. An example of search engine *Bing* (https://cn.bing.com/) with machine reading comprehension (MRC) techniques.

Due to the small size of human-generated datasets and the limitations of rule-based and machine-learning-based methods, early MRC systems did not perform well and hence could not be used in practical applications. This situation has changed since 2015, which can be attributed to two driving forces. On the one hand, MRC based on deep learning, also called neural machine reading comprehension, shows its superiority in capturing contextual information and dramatically outperforms traditional systems. On the other hand, a variety of large-scale benchmark datasets, such as CNN & Daily Mail [5], Stanford Question-Answering Dataset (SQuAD) [6] and MS MARCO [7], make it possible to solve MRC tasks with deep neural architectures and provide testbeds to extensively evaluate the performance of MRC systems. To illustrate the development trends of neural MRC more clearly, we conducted a statistical analysis of representative articles in this field, and the result is presented in Figure 2. As shown in this figure, on the whole, the number of articles increased exponentially from 2015 to the end of 2018. Furthermore, as time goes on, the types of MRC tasks are becoming increasingly diverse. All these demonstrate that neural MRC is under rapid development and has become the research focus of both academia (Stanford, Carnegie Mellon, etc.) and industry (Google, Facebook, Microsoft, etc.) demonstrated by a large number of authors coming from these institutions.

The flourishing research of neural MRC calls for surveys to systemically study and analyze the recent successes. However, such a comprehensive review still cannot be found. Though Qiu et al. [8] recently give a brief overview to illustrate how to use deep-learning methods to deal with MRC tasks by introducing several classic neural MRC models, they neither give specific definitions of different MRC tasks nor compare them in depth. Moreover, they do not discuss the new trends and open issues in this field. Motivated by the lack of published surveys, we conducted a thorough literature review of recent progress in neural MRC with the expectation that it would help researchers, in particular newcomers, to obtain a panoramic view of this field. To achieve that goal, we collected papers mainly using Google Scholar (http://scholar.google.com) with keywords including machine reading comprehension, machine comprehension, reading comprehension, deep learning, and neural networks. We selected from the search results only papers published on related high-profile conferences such as

ACL, EMNLP, NAACL, ICLR, AAAI, IJCAI and CoNLL, and restricted the time range to 2015–2018. In addition, arXiv (http://arxiv.org/), which includes some of the latest pre-print articles, was used as a supplementary source. Based on more than 85 frequently cited papers collected, a possible organization of the various facets of MRC as a field is shown in Figure 3. Our article follows a similar structure. First, we group common MRC tasks into four types according to Chen's categorization in her PhD thesis [9]: *cloze tests*, *multiple choice*, *span extraction*, and *free answering*. We further extend this taxonomy by giving a formal definition to each type, comparing these tasks in different dimensions (Section 2). Second, we present the general architecture of neural MRC systems, which consists of four modules: *embeddings*, *feature extraction*, *context-question interaction* and *answer prediction*. The prevalent deep learning techniques used in each module are also detailed (Section 3). Third, some representative datasets and evaluation metrics are described according to different tasks (Section 4). Fourth, some new trends, such as *knowledge-based MRC*, *MRC with unanswerable questions*, *multi-passage MRC* and *conversational MRC*, are revealed by figuring out their challenges and describing existing approaches and limitations (Section 5). Finally, several open issues are discussed with the hope of shedding light on possible future research directions (Section 6). Before proceeding on the next section, as there are lots of terms mentioned in this article, to improve the readability for newcomers, definitions of terms can be found in a glossary presented in Appendix A.

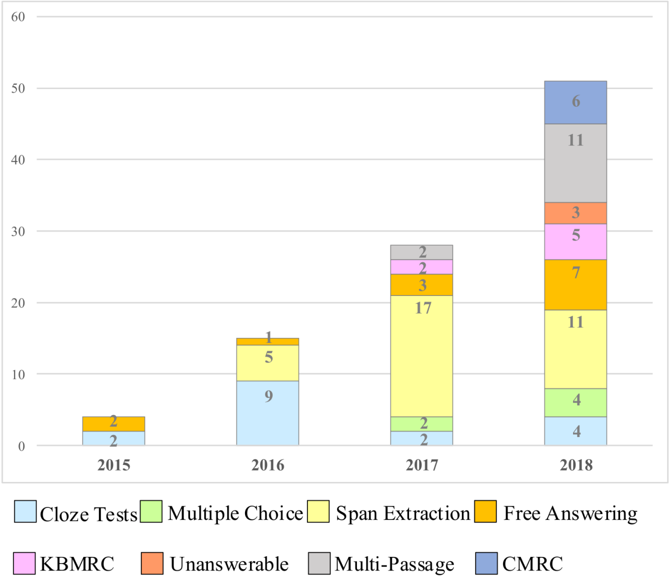


Figure 2. The number of research articles concerned with neural MRC in this survey. KBMRC: knowledge-based machine reading comprehension; CMRC: conversational machine reading comprehension.

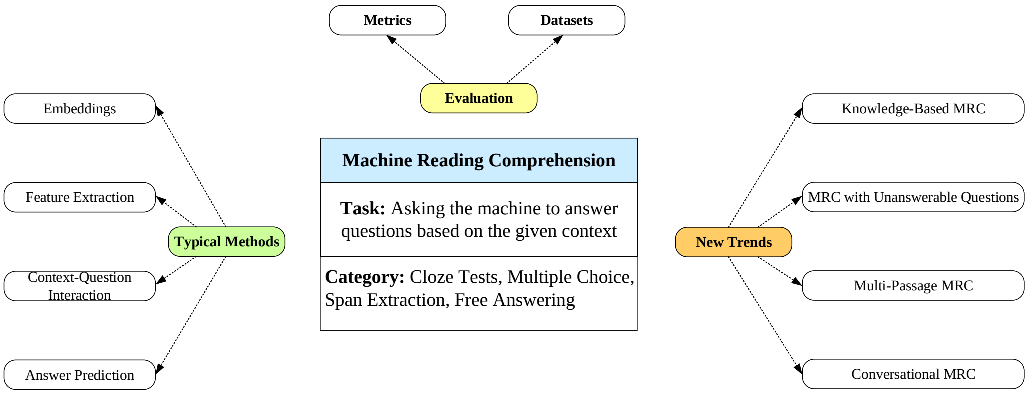

Metrics
Datasets
Evaluation
Embeddings
Feature Extraction
Typical Methods
Context-Question Interaction
Answer Prediction
Machine Reading Comprehension
Task: Asking the machine to answer questions based on the given context
Category: Cloze Tests, Multiple Choice, Span Extraction, Free Answering
New Trends
Knowledge-Based MRC
MRC with Unanswerable Questions
Multi-Passage MRC
Conversational MRC

Figure 3. Facets of machine reading comprehension reflected in the structure of this article.

2. Tasks

Machine reading comprehension (MRC) is a basic task of textual question answering (QA), in which each question is given related context from which to infer the answer. The objective of MRC is to extract the correct answer from the given context or even generate a more complex answer based on the context. MRC holds the promise of bridging the gap of understanding natural language between humans and machines. The formal definition of MRC is as shown in Table 1:

Table 1. Definition of machine reading comprehension.

Machine Reading Comprehension
Given the context C and question Q, machine reading comprehension tasks ask the model to give the correct answer A to the question Q by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C, Q)$.

In this section, following Chen [9], we categorize MRC into four tasks, mainly based on the answer form: cloze tests, multiple choice, span extraction and free answering. For better understanding, some examples of representative datasets are presented in Table 2. In the following part, we give a description of each task with a formal definition and compare them from different dimensions.

Table 2. A few examples of MRC datasets

Cloze Tests		
CNN & Daily Mail [5]	Context:	the *ent381* producer allegedly struck by *ent212* will not press charges against the "*ent153*" host, his lawyer said Friday. *ent212*, who hosted one of the most-watched television shows in the world, was dropped by the *ent381* Wednesday after an internal investigation by the *ent180* broadcaster found he had subjected producer *ent193* "to an unprovoked physical and verbal attack."
	Question:	producer **X** will not press charges against *ent212*, his lawyer says.
	Answer:	*ent193*
Multiple Choice		
RACE [10]	Context:	If you have a cold or flu, you must always deal with used tissues carefully. Do not leave dirty tissues on your desk or on the floor. Someone else must pick these up and viruses could be passed on.
	Question:	Dealing with used tissues properly is important because _____.
	Options:	A. it helps keep your classroom tidy B. people hate picking up dirty tissues C. it prevents the spread of colds and flu D. picking up lots of tissues is hard work
	Answer:	C
Span Extraction		
SQuAD [6]	Context:	Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.
	Question:	By what main attribute are computational problems classified using computational complexity theory?
	Answer:	inherent difficulty
Free Answering		
MS MARCO [7]	Context 1:	Rachel Carson's essay on The Obligation to Endure, is a very convincing argument about the harmful uses of chemical, pesticides, herbicides and fertilizers on the environment.
	......	
	Context 5:	Carson believes that as man tries to eliminate unwanted insects and weeds; however he is actually causing more problems by polluting the . environment with, for example, DDT and harming living things
	......	
	Context 10:	Carson subtly defers her writing in just the right writing for it to not be subject to an induction run rampant style which grabs the readers interest without biasing the whole article.
	Question:	Why did Rachel Carson write an obligation to endure?
	Answer:	Rachel Carson writes The Obligation to Endure because believes that as man tries to eliminate unwanted insects and weeds; however he is actu -ally causing more problems by polluting the environment.

2.1. Cloze Tests

Cloze tests, also known as gap-filling tests, are commonly adopted in exams to evaluate students' language proficiency. Inspired by that, this task is used to measure the ability of machines to understand natural language. In cloze tests, questions are generated by removing some words or entities from a passage. To answer questions, one is asked to fill in the blank with the missing items. Some tasks provide candidate answers, but this is optional. Cloze tests, which add obstacles to reading, require understanding of context as well as usage of vocabulary and are challenging for machine reading comprehension.

The features of cloze tests are as follows: (i) answer A is a word or entity in the given context C; and (ii) question Q is generated by removing a word or entity from the given context C such that $Q = C - A$. According to such features, cloze tests can be formulated as shown in Table 3:

Table 3. Definition of cloze tests.

Cloze Tests
Given the context C, from which a word or an entity $A(A \in C)$ is removed, the cloze tests ask the model to fill in the blank with the right word or entity A by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C - \{A\})$.

2.2. Multiple Choice

Multiple choice is another machine reading comprehension task inspired by language proficiency exams. It requires selecting the correct answer to the question from candidates according to the provided context. Compared to cloze tests, answers for multiple choice are not limited to words or entities in the context, so the answer form is more flexible, but it is required for this task to provide candidate answers.

The feature of multiple choice is that a list of candidate answers $A = \{A_1, A_2, \cdots, A_n\}$ is given that can act as auxiliary information to predict the answer. It can be formulated as shown in Table 4:

Table 4. Definition of multiple choice.

Multiple Choice
Given the context C, the question Q and a list of candidate answers $A = \{A_1, A_2, \cdots, A_n\}$, the multiple-choice task is to select the correct answer A_i from A $(A_i \in A)$ by learning the function $\mathcal{F}$ such that $A_i = \mathcal{F}(C, Q, A)$

2.3. Span Extraction

Although cloze tests and multiple choice can measure the machine's ability to understand natural language to some extent, there are limitations in those tasks. More specifically, words or entities are not sufficient to answer questions. Instead, some complete sentences are required. Moreover, there are no candidate answers in many cases. The span extraction task can overcome these weaknesses. Given the context and the question, this task asks the machine to extract a span of text from the corresponding context as the answer.

The feature of span extraction is that answer A is required to be a continuous subsequence of the given context C. It can be formulated as shown in Table 5:

Table 5. Definition of span extraction.

Span Extraction
Given the context C, which consists of n tokens, that is $C = \{t_1, t_2, \cdots, t_n\}$, and the question Q, the span extraction task requires extracting the continuous subsequence $A = \{t_i, t_{i+1}, \cdots, t_{i+k}\}(1 \leq i \leq i+k \leq n)$ from context C as the correct answer to question Q by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C, Q)$.

2.4. Free Answering

Compared to cloze tests and multiple choice, the span extraction task makes great strides in allowing machines to give more flexible answers, yet it is not enough, because giving answers restricted to a span of the context is still unrealistic. To answer the questions, the machine needs to reason across multiple pieces of the text and summarize the evidence. Among the four tasks, free answering is the most complicated, as there are no limitations to its answer forms, and it is more suitable for real application scenarios.

In contrast to the other tasks, free answering reduces some constraints and focuses much more on using free-form natural language to better answer questions and it can be formulated as shown in Table 6:

Table 6. Definition of free answering.

Free Answering
Given the context C and the question Q, the correct answer A in free answering task may not be subsequence in the original context C, namely either $A \subseteq C$ or $A \nsubseteq C$. The task requires predicting the correct answer A by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C, Q)$.

2.5. Comparison of Different Tasks

To compare the contribution and limitations of the four MRC tasks, we evaluated five dimensions: construction, understanding, flexibility, evaluation and application. In each dimension, we compare tasks and score them according to relative ranking. As there are four tasks under consideration, score 4 means the corresponding task has the best performance in that dimension while score 1 means it performs the worst. Moreover, two tasks score similarly means that it is hard to judge which one is better in the dimension.

- Construction: This dimension measures whether it is easy to construct datasets for the task or not. The easier it is, the higher the score.
- Understanding: This dimension evaluates how well the task can test the machine's ability to understand. If a task needs more understanding and reasoning, the score is higher.
- Flexibility: The flexibility of the answer form can measure the quality of the tasks. When answers are more flexible, the flexibility score is higher.
- Evaluation: Evaluation is a necessary part of MRC tasks. Whether a task can be easily evaluated also determines its quality. Tasks that are easy to evaluate get high scores in this dimension.
- Application: A good task is supposed to be close to real-world application. Therefore, scores in this dimension are high, if a task can easily be applied to the real world.

As presented in Figure 4, scores of these five dimensions vary from different tasks. More specifically, it is easiest to construct datasets and evaluate cloze tests. However, as the answer form is restricted to a single word or name entity in the original context, cloze tests cannot test machine understanding well and do not conform with real-world applications. Multiple-choice tasks provide candidate answers for each question, so that even if answers are not limited in the original context, they can be easily evaluated. It is not very hard to build datasets for this task as multiple-choice tests in language exams can be easily used. However, candidate answers lead to a gap between synthetic datasets and realistic application. In contrast, span extraction tasks are a moderate choice, for which datasets can easily be constructed and evaluated. Moreover, they can test a machine's understanding of text, in a way. All these advantages contribute to quite a lot research focusing on these tasks. The disadvantage of span extraction is that answers are constrained to the subsequence of original context, which is still a little far from the real world. Free answering tasks show their superiority in the dimensions of understanding, flexibility, and application, which are the closest to practical application. However, every coin has two sides. Because of the flexibility of the answer form, it is somewhat hard to build datasets, and how to effectively evaluate performance on these tasks remains a challenge.

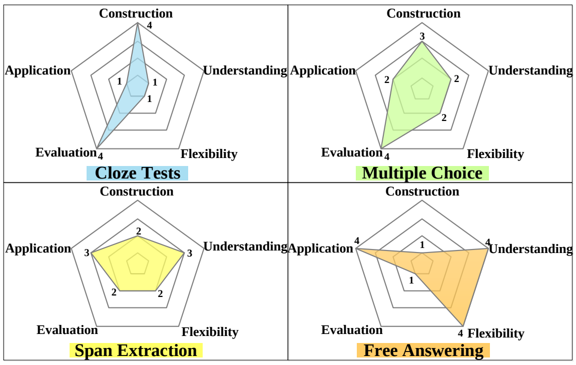


Figure 4. Comparison of different MRC tasks.

3. Deep-Learning-Based Methods

With the release of the CNN & Daily Mail dataset [5] and the development of deep-learning techniques, neural MRC shows superiority over traditional rule-based and machine-learning-based MRC, and has gradually become the mainstream in the research community. In this section, we introduce the general architecture of neural MRC systems, followed by an illustration of typical deep-learning methods applied to different modules.

3.1. General Architecture

As presented in Figure 5, a typical neural machine reading comprehension system, which takes the context and question as inputs and the answer as output, contains four key modules: embeddings, feature extraction, context-question interaction and answer prediction. The function of each module can be interpreted as follows.

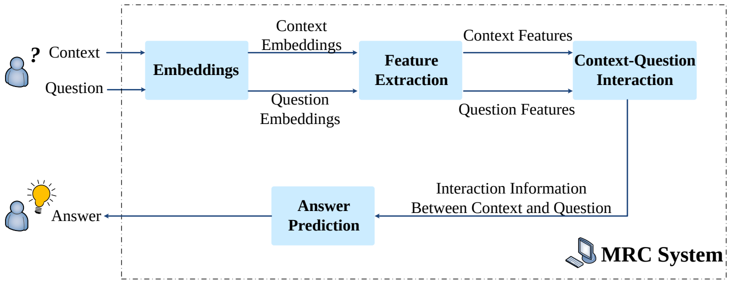


Figure 5. The general architecture of machine reading comprehension system.

- Embeddings: As the machine is unable to understand natural language directly, the embedding module is indispensable to change input words into fixed-length vectors at the beginning of the MRC systems. Taking the context and question as inputs, this module outputs context and question embeddings by various approaches. Classical word representation methods such as one-hot and Word2Vec, sometimes combined with other linguistic features, i.e., part-of-speech, name entity, and question category, are usually used to represent semantic and syntactic information in the words. Moreover, contextualized word representations pre-trained by a large corpus also show promising performance in encoding contextual information.
- Feature Extraction: After the embedding module, context and question embeddings are fed to the feature extraction module. To better understand the context and question, this module is aimed at extracting more contextual information. Some typical deep neural networks, such as recurrent neural networks (RNNs) and convolution neural networks (CNNs) are applied to further mine contextual features from context and question embeddings.
- Context-Question Interaction: The correlation between the context and the question plays a significant role in predicting the answer. With such information, the machine can find out which

parts in the context are more important to answering the question. To achieve that goal, the attention mechanism, unidirectional or bidirectional, is widely used in this module to emphasize parts of the context relevant to the query. To sufficiently extract their correlation, the interaction between the context and the question sometimes involves multiple hops, which simulates the rereading process of human comprehension.

- Answer Prediction: Answer prediction module, the last component of an MRC system, outputs the final answer based on all the information accumulated from previous modules. As MRC tasks can be categorized according to answer forms, this module is highly related to different tasks. For cloze tests, the output of this module is a word or entity in the original context, while the multiple-choice task requires selecting the correct answer from candidate answers. In terms of span extraction, this module extracts a subsequence of the given context as the answer. Some generation techniques are used in this module for the free answering task, as it has nearly no constraint on answer forms.

3.2. Typical Deep-Learning Methods

Compared to traditional rule-based and machine-learning-based methods, deep-learning techniques show their superiority in extracting contextual information, which is very important to MRC tasks. In this section, we present various deep-learning approaches used in different modules of MRC systems in Figure 6 and summarize those approaches applied in classic MRC models in Table 7. Moreover, some tricks to improve the performance of MRC systems are introduced in the last part of this section.

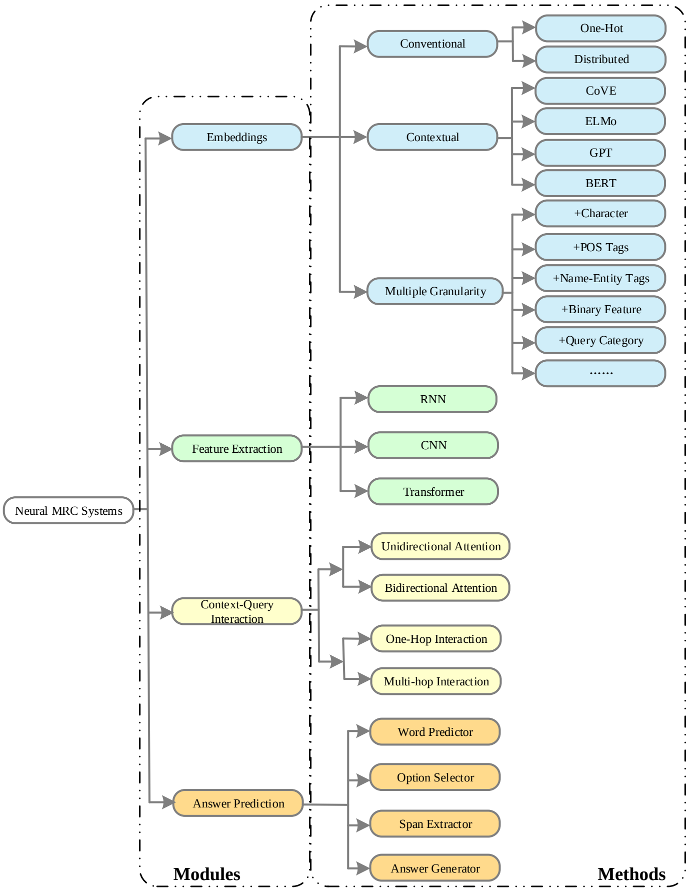


Figure 6. Typical techniques in neural MRC systems.

Table 7. Classic neural MRC models (C: Cloze Tests, M: Multiple Choice, S: Span Extraction, F: Free Answering).

Models	Methods			
	Embeddings	**Feature Extraction**	**Context-Question Interaction**	**Answer Prediction**
Attentive Reader (Hermann et al. 2015) [5]	Conventional	RNN	Unidirectional One-Hop	Word Predictor
Impatient Reader (Hermann et al. 2015) [5]	Conventional	RNN	Unidirectional Multi-hop	Word Predictor
End-to-End Memory Networks (Sukhbaatar et al. 2015) [11]	Conventional	/	Multi-Hop	Word Predictor
Standford Reader (Chen et al. 2016) [12]	Conventional	RNN	Unidirectional One-Hop	Word Predictor
AS Reader (Kadlec et al. 2016) [13]	Conventional	RNN	Unidirectional One-Hop	Word Predictor
IA Reader (Sordoni et al. 2016) [14]	Conventional	RNN	Unidirectional Multi-Hop	Word Predictor
Match-LSTM (Wang & Jiang 2016) [15]	Conventional	RNN	Unidirectional Multi-Hop	Span Extractor
Bi-DAF (Seo et al. 2016) [16]	Multiple Granularity	RNN	Bidirectional One-Hop	Span Extractor
DCN (Xiong et al. 2016) [17]	Conventional	RNN	Bidirectional One-Hop	Span Extractor
AoA Reader (Cui et al. 2017) [18]	Conventional	RNN	Bidirectional One-hop	Word Predictor
GA Reader (Dhingta et al. 2017) [19]	Conventional	RNN	Unidirectional Multi-Hop	Word Predictor
MEMEN (Pan et al. 2017) [20]	Multiple Granularity	RNN	Multi-Hop	Span Extractor
R-NET (Wang et al. 2017) [21]	Multiple Granularity	RNN	Unidirectional Multi-Hop	Span Extractor
DCN+ (Xiong et al. 2017) [22]	Conventional	RNN	Bidirectional Multi-Hop	Span Extractor
S-NET (Tan et al. 2017) [23]	Multiple Granularity	RNN	Unidirectional Multi-Hop	Answer Generator
CoVe + DCN (Mccann et al. 2017) [24]	Contextual	RNN	Bidirectional One-Hop	Span Extractor
QANet (Yu et al. 2018) [25]	Multiple Granularity	Transformer	Bidirectional One-Hop	Span Extractor
Reinforced Mnemonic Reader (Hu et al. 2018) [26]	Multiple Granularity	RNN	Bidirectional Multi-Hop	Span Extractor
CSA (Chen et al. 2018) [27]	Multiple Granularity	RNN&CNN	Bidirectional One-Hop	Option Selector
CNN Model (Chaturvedi et al. 2018) [28]	Conventional	CNN	Unidirectional One-Hop	Option Selector
Hierarchical Attention Flow (Zhu et al. 2018) [29]	Conventional	RNN	Bidirectional One-Hop	Option Selector
ELMo + improved Bi-DAF (Peters et al. 2018) [30]	Contextual	RNN	Bidirectional One-Hop	Span Extractor
GPT (Radford et al. 2018)[31]	Contextual	/	/	Option Selector
BERT (Devlin et al. 2018) [32]	Contextual	/	/	Span Extractor

3.2.1. Embeddings

The embedding module is an essential part of an MRC systems and is usually placed at the beginning to encode input natural language words into fixed-length vectors, which the machine can understand and deal with. As Dhingra et al. [33] point out, minor choices made in word representation can lead to substantial differences in the final performance of the reader. How to sufficiently encode the context and question is the pivotal task of this module. In existing MRC models, word representation methods can be sorted into conventional word representation and pre-trained contextualized representation. To encode more abundant semantic and linguistic information, multiple granularity, which fuses word-level embeddings with character-level embeddings, part-of-speech, name entity, word frequency, question category, and so on, is also applied to some MRC systems. In the following parts, we will give a detailed illustration.

(1) Conventional Word Representation

- One-Hot

This method [34] represents words with binary vectors, and its size is same as the number of words in the dictionary. In such vectors, one position is 1, corresponding to the word, while all others are 0. As a word representation approach at the early stage, it can encode words when the size of the vocabulary is not very large. However, this representation is sparse and may suffer from the curse of dimensionality with increased vocabulary. In addition, one-hot encoding cannot represent relationships among words. For instance, "apple" and "pear" belong to the category "fruit", but their word representations embedded by one-hot cannot show such a relationship.

- Distributed Word Representation

To address the shortcomings of representations such as one-hot, Rumelhart et al. [35] propose distributed word representation, which encodes words into continuous low-dimensional vectors. Closely related words encoded by these methods are not far from each other in vector space, which reveals the correlation of words. Various techniques to generate distributed word representations have been introduced, among which the most popular are Word2Vec [36] and GloVe [37]. Besides successful applications in a variety of Natural Language Processing (NLP) tasks, such as machine translation [38] and sentiment analysis [39], vectors produced by these methods are also applied to many MRC systems.

(2) Pre-trained Contextualized Word Representation

Although distributed word representation can encode words in low-dimensional space and reflect correlations between words, it cannot efficiently mine contextual information. To be specific, a vector produced by the distributed word representation of one word is constant regardless of different contexts. To address this problem, researchers have introduced contextualized word representations, which are pre-trained with large corpora in advance and then directly used as conventional word representations or fine-tuned according to the specific tasks. This is a kind of transfer learning and has shown promising performance in a wide range of NLP tasks including machine reading comprehension. Even a simple neural network model can perform well in answer prediction with such a pre-trained word representation approach.

- CoVE

Inspired by successful cases in computer vision, which transfer CNNs pre-trained on a large supervised training corpus such as ImageNet to other tasks, McCann et al. [24] try to bring the benefit of transfer learning to NLP tasks. They first train long short-term memory (LSTM) encoders of sequence-to-sequence models on a large-scale English-to-German translation dataset and then transfer the outputs of the encoder to other NLP tasks. As machine translation (MT) requires the model to encode words in context, the output of the encoder can be regarded as context vectors (CoVE). To deal with MRC problems, McCann et al. concatenate the outputs of an MT encoder with word embeddings pre-trained by GloVe to represent the context and question and feed them through the coattention and dynamic decoder implemented in a dynamic coattention network (DCN) [17]. DCN with CoVe outperforms the original DCN on the SQuAD dataset, which illustrates the contribution of contextualized word representations to downstream tasks. However, pre-training CoVE requires a large parallel corpus. Its performance will degrade if the training corpus is not adequate.

- ELMo

Embeddings from language models (ELMo), proposed by Peters et al. [30], is another type of contextualized word representation. To get ELMo embeddings, they first pre-train a bidirectional language model (biLM) with a large text corpus. Compared to CoVe, ELMo breaks the constraint of limited parallel corpus and obtains richer word representations by collapsing outputs of all biLM layers into a single vector with a task-specific weighting rather than using outputs of the top layer. Model evaluations illustrate that different levels of LSTM states can capture diverse syntactic and linguistic information. When applying ELMo embeddings to MRC models, Peters et al. choose an improved version of bidirectional attention flow (Bi-DAF), introduced by Clark and Gardner [40], as the baseline and improved the state-of-the-art single model by 1.4% on the SQuAD dataset. ELMo, which can be easily integrated with existing models, shows promising performance on various NLP tasks, but it is limited by the insufficient feature extraction capability of LSTM.

- GPT

Generative pre-training (GPT) [31], is a semi-supervised approach combining unsupervised pre-training and supervised fine-tuning. Representations pre-trained by this method can transfer to various NLP tasks with little adaptation. The basic component of GPT is a multi-layer transformer [41] decoder that mainly uses multi-head self-attention to train the language model and allow longer semantic structure to be captured compared to RNN-based models. After training, the pre-trained parameters are fine-tuned for specific downstream tasks. In terms of MRC problems such as multiple choice, Radford et al. concatenate the context and the question with each possible answer and process such sequences with transformer networks. Finally, they produce an output distribution over possible answers to predict correct answers. GPT achieves an improvement of 5.7% on the RACE [10] dataset compared with the state-of-the-art. Seeing the benefit brought by contextualized word representations pre-trained on large-scale datasets, Radford et al. [42] later propose GPT-2, which is pre-trained on a larger corpus, WebText, with more than 1.5 billion parameters. Compared to the previous model, the layers of transformer architecture are increased from 12 to 48. Moreover, single task training is substituted with a multi-task learning framework, which makes GPT-2 more generative. This improved version shows competitive performance even in zero-shot setting. However, the transformer architecture used in GPT and GPT-2 is unidirectional (left-to-right), which cannot incorporate context from both directions. This may be the major shortcoming and limits its performance on downstream tasks.

- BERT

Considering the limitations of unidirectional architecture applied in previous pre-training models such as GPT, Devlin et al. [32] propose a new model, bidirectional encoder representation from transformers (BERT). With the masked language model (MLM) and next-sentence prediction task, BERT can pre-train deep contextualized representations with a bidirectional transformer, encoding both left and right context for word representations. As transformer architecture cannot extract sequential information, Devlin et al. add positional embeddings to encode position. Owing to the bidirectional language model and transformer architecture, BERT outperforms state-of-the-art models on 11 NLP tasks. In particular, for MRC tasks, BERT is so competitive that using it with a simple answer prediction approach shows promise. Despite its outstanding performance, BERT's pre-training process is time and resource-consuming which makes it nearly impossible to pre-train without abundant computational resources.

(3) Multiple Granularity

Word-level embeddings pre-trained by Word2Vec or GloVe cannot encode rich syntactic and linguistic information, such as part-of-speech, affixes, and grammar, which may not be sufficient for deep machine understanding. To incorporate fine-grained semantic information into word representations, some researchers have introduced approaches to encode the context and the question at different levels of granularity.

- Character Embeddings

Character embeddings represent words at the character level. Compared to word-level representations, they are not only more suitable for modeling sub-word morphologies but also can alleviate the out-of-vocabulary (OOV) problem. Seo et al. [16] add character-level embeddings to their Bi-DAF model for the MRC task. They use CNNs to obtain character-level embeddings. Each character in the word is embedded into a fixed-dimension vector, which is fed to CNNs as 1D inputs. After max-pooling the entire width, the outputs of CNNs are embeddings at the character level. The concatenation of word-level and character-level embeddings are then fed to the next module as input. In addition, character embeddings can be encoded with bidirectional LSTMs [26,43]. For each word, the outputs of the last hidden state are considered to be its character-level representation. Moreover, word-level and character-level embeddings can be combined dynamically with a fine-grained gating mechanism rather than simple concatenation to mitigate the imbalance between frequent and infrequent words [44].

- Part-of-Speech Tags

A part-of-speech (POS) is a particular grammatical class of words, such as nouns, adjectives, or verb. Labeling POS tags in NLP tasks can illustrate complex characteristics of word use and in turn contribute to disambiguation. To translate POS tags into fixed-length vectors, they are regarded as variables, randomly initialized in the beginning and updated while training.

- Name-Entity Tags

Name entity, a concept in information retrieval, refers to a real-world object, such as a person, location, or organizations, with a proper name. When asking about such objects, name entities are probable answer candidates. Thus, embedding name-entity tags of context words can improve the accuracy of answer prediction. The method of encoding name-entity tags is similar to that of POS tags mentioned above.

- Binary Feature of Exact Match (EM)

This feature, which measures whether a context word is in the question, was first used in the conventional entity-centric model proposed by Chen et al. [45]. Later, some researchers used it in the Embedding module to enrich word representations. The value of this binary is 1 if a context word can be exactly matched to one word in the query, otherwise its value is 0. More loosely, Chen et al. [27] use partial matching to measure the correlation between context words and question words. For instance, "teacher" can be partially matched with "teach".

- Query-Category

The types of questions (what, where, who, when, why, how) can usually provide clues to search for the answer. For instance, a "where" question pays more attention to spatial information. Zhang et al. [46] introduce a method to model different question categories in end-to-end training. They first obtain query types by counting the key word frequency. Then the question type information is encoded to one-hot vectors and stored in a table. For each query, they look up the table and use a feed-forward neural network for projection. The query-category embeddings are often added to the query word embeddings.

The embeddings introduced above can be combined freely in the embedding module. Hu et al. [26] use word-level, character-level, POS tags, name-entity tags, binary feature of EM, and query-category embeddings in their Reinforced Mnemonic Reader to incorporate syntactic and linguistic information in word representations. Experimental results show that rich word representations contribute to deep understanding and improve answer prediction accuracy.

To sum up, word embeddings encoded by distributed word representations are the basis of this module. As more abundant representations with syntactic and linguistic information contribute to better performance, multiple granularity representations have gradually become prevalent. In terms of contextual word representation, they can improve performance dramatically, and can be used by themselves or combined with other representations.

3.2.2. Feature Extraction

The feature extraction module is often placed after the embedding layer to extract features of the context and question separately. It further pays attention to mining contextual information at the sentence level based on various types of syntactic and linguistic information encoded by the embedding module. Recurrent neural networks (RNNs), convolution neural networks (CNNs) and Transformer architecture are applied in this module, and we will give an illustration in detail in this part.

(1) Recurrent Neural Networks

RNNs are popular models that have shown great promise for dealing with sequential information. RNNs are called *recurrent* as outputs in each time step depending on the previous computations. RNN-based models have been widely used in various NLP tasks, such as machine translation, sequence tagging and question answering. In particular, long short-term memory (LSTM) [47] networks and gated recurrent units (GRUs) [38], variants of RNNs, are much better at capturing long-term dependencies than plain ones are and can alleviate gradient explosion and vanishing problems. Since the preceding and following words have the same importance in understanding the given word, many researchers use bidirectional RNNs to encode the context and question embeddings in MRC systems. The context and question embeddings are denoted as x_p and x_q, respectively, and we will illustrate how the feature extraction module with bidirectional RNNs handles those embeddings and extracts sequential information.

In terms of questions, the feature extraction process with bidirectional RNNs can be sorted into two types: word-level and sentence-level.

In word-level encoding, feature extraction outputs for each question embedding x_{qj} at time step j can be denoted as follows:

$$Q_j = [\overrightarrow{RNN}(x_{qj}); \overleftarrow{RNN}(x_{qj})], \tag{1}$$

where $\overrightarrow{RNN}(q_{xj})$ and $\overleftarrow{RNN}(q_{xj})$ denotes forward and backward hidden states of bidirectional RNNs, respectively. This process is shown in Figure 7 in detail.

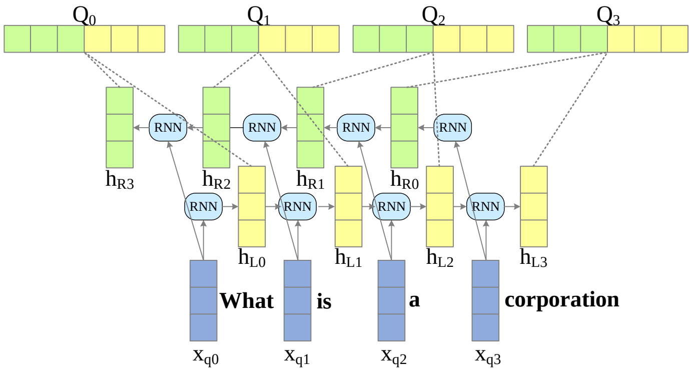


Figure 7. Word-level encoding for questions.

By contrast, sentence-level encoding regards the question sentence as a whole. The feature extraction process can be denoted as:

$$Q = [\overrightarrow{RNN}(x_{q|l|}); \overleftarrow{RNN}(x_{q0})], \tag{2}$$

where $|l|$ is the length of the question, $\overrightarrow{RNN}(x_{q|l|})$ and $\overleftarrow{RNN}(x_{q0})$ represent final forward and backward outputs of RNNs, respectively. To be more concrete, we demonstrate this sentence-level encoding process in Figure 8.

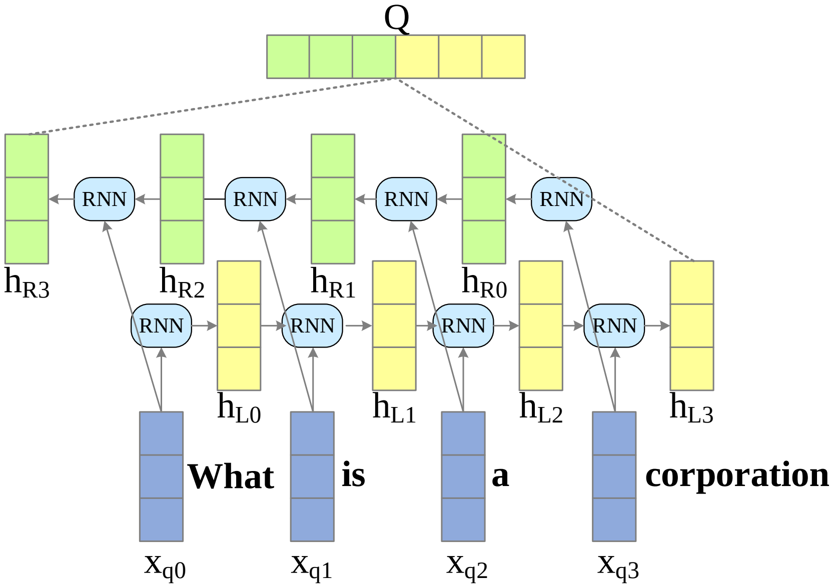


Figure 8. Sentence-level encoding for questions.

As the context in an MRC task is usually a long sequence, researchers use word-level feature extraction to encode sequential information of context. Similar to question encoding, the feature extraction process with bidirectional RNNs for context embedding x_{pi} at time step i can be denoted as:

$$P_i = [\overrightarrow{RNN}(x_{pi}); \overleftarrow{RNN}(x_{pi})]. \tag{3}$$

Although RNNs are capable of modeling sequential information, their training process is time-consuming as they cannot be processed in parallel.

(2) Convolution Neural Networks

CNNs are widely used in computer vision. When applied to NLP tasks, one-dimensional CNNs show their superiority in mining local contextual information with sliding windows. In CNNs, each convolutional layer applies a different scale of feature maps to extract local features in diverse window sizes. The outputs are then fed to pooling layers to reduce dimensionality but keep the most significant information to the greatest extent. Maximum and average operations on the results of each filter are common ways to do pooling. Figure 9 shows how the feature extraction module uses CNNs to mine the local contextual information of a question.

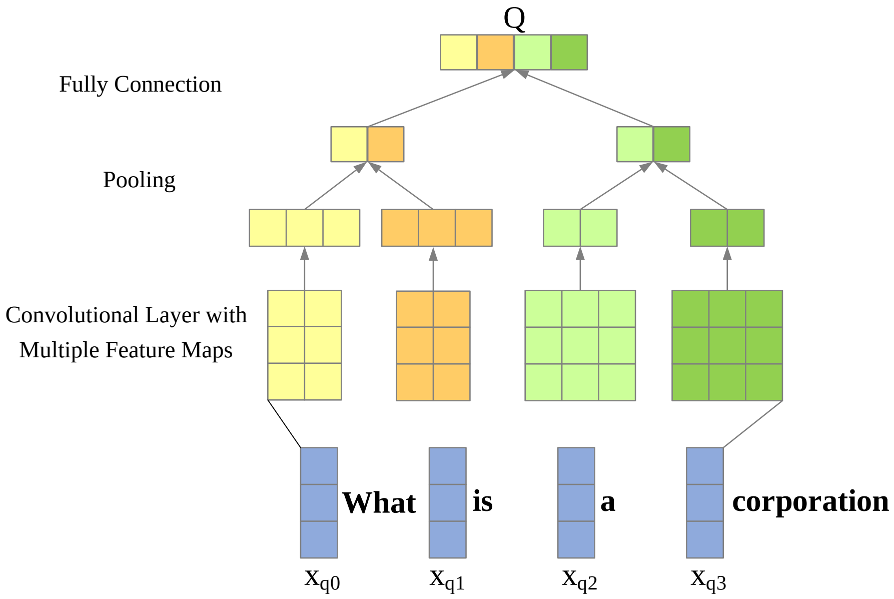


Figure 9. Using CNNs to extract features of question.

As shown in Figure 9, given word embeddings of a question $x_q \in \mathbb{R}^{|l| \times d}$, where $|l|$ represents the length of the question and d denotes the dimension of word embeddings, the convolution layer has two filter sizes $f_t \times d(\forall t = 2,3)$ with k output channels ($k = 2$ in the example in Figure 9). Each filter produces a feature map of shape $(|l| - t + 1) \times k$ (padding is 0 and stride is 1), which are pooled to generate a k-dimensional vector. The two k-dimensional vectors are concatenated to form a 2k-dimensional vector, represented by Q.

Although both n-gram models and CNNs can focus on local features of the sentence, training parameters in n-gram models increase exponentially with larger vocabularies. By contrast, CNNs can extract local information in a more compact and efficient way regardless of the vocabulary size, as there is no need for them to represent every n-gram in the vocabulary. In addition, CNNs can be trained in parallel, which is faster than RNNs. One major shortcoming of CNNs is that they can extract only local information but are not capable of dealing with long sequence.

(3) Transformer

The transformer, proposed by Vaswani et al. [41] in 2017, is a powerful neural network model that has shown promising performance on various NLP tasks [31,32]. In contrast to RNN-based or CNN-based models, the transformer is mainly based on the attention mechanism with neither recurrence nor convolution. Owing to multi-head self-attention, this simple architecture not only excels in alignment but also can be run in parallel. Compared to RNNs, the transformer requires less time to train, while it pays more attention to global dependencies, in contrast to CNNs. However, without recurrence and convolution, the model cannot make use of the order of the sequence. To incorporate positional information, Vaswani et al. add position encoding computed by sine and cosine functions. The sum of positional and word embeddings is fed to the transformer as inputs. Figure 10 presents the simple architecture of the transformer. In practice, models usually stack several blocks with multi-head self-attention and feed-forward network.

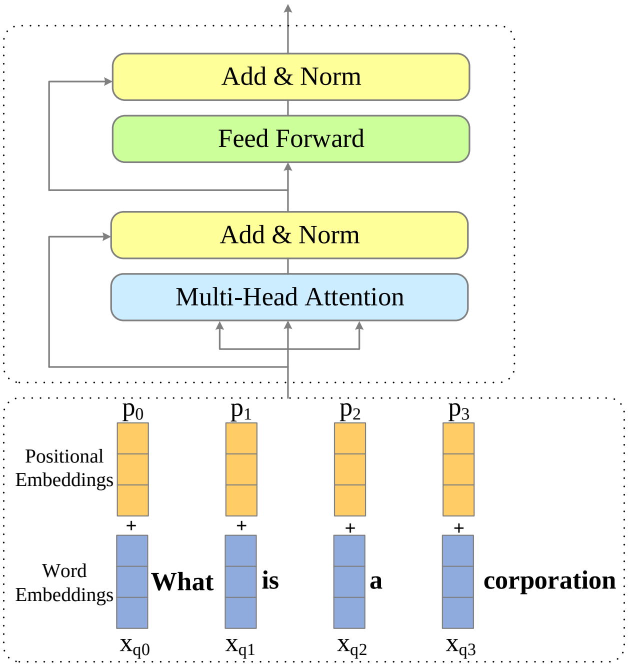


Figure 10. Using the Transformer to extract features of question.

QANet, introduced by Yu et al. [25], is a representative MRC model that uses the transformer. The basic encoder block of QANet is a novel architecture, which combines the multi-head self-attention defined in the transformer with convolutions. The experimental results show that QANet achieves the same accuracy on SQuAD as the prevalent recurrent models with much faster training and inference speed.

In general, most MRC systems use RNNs in the feature extraction module because of their superiority in handling sequential information. To accelerate the training process, some researchers substitute RNNs with CNNs or the transformer. CNNs are highly parallelized and can obtain rich local information with feature maps in different sizes. The transformer can mitigate the side-effect of long dependency and improve computational efficiency.

3.2.3. Context-Question Interaction

By extracting the correlation between the context and the question, models can find evidence for answer prediction. Inspired by Hu et al. [26], we can divide existing works into two kinds according to how models extract correlations: one-hop and multi-hop interaction. No matter what kind of interaction the MRC model uses, the attention mechanism plays a critical role in emphasizing which parts of the context are more important to answer the questions.

Derived from human intuition, the attention mechanism was adapted for machine translation and shows promising performance on automatic token alignment [48,49]. As a simple and effective method that can be used to encode sequence data with its importance, it has shown significant improvement in various tasks in natural language processing, including text summarization [50], sentiment classification [51], semantic parsing [52], etc. In terms of machine reading comprehension, the attention mechanism can be categorized into unidirectional and bidirectional attention according to whether it is used unidirectionally or bidirectionally. In the following part, we introduce methods categorized using the attention mechanism, followed by illustrations of one-hop and multi-hop interactions.

(1) Unidirectional Attention

Unidirectional attention flow is usually from query to context, highlighting the most relevant parts of the context according to the question. It is believed that if the context word is more similar to the question, it is more likely to be the answer word. As shown in Figure 11, the similarity of each context semantic embedding P_i and the whole question sentence representations Q (by sentence-level encoding introduced in Section 3.2.2) is calculated by $S_i = f(P_i, Q)$, where $f(\cdot)$ represents the function that can measure the similarity. After being normalized by the SoftMax function in Equation (4), the attention weight α_i for each context word is obtained, with which the MRC systems can finally predict the answer.

$$\alpha_i = \frac{\exp S_i}{\sum_j \exp S_j}. \tag{4}$$

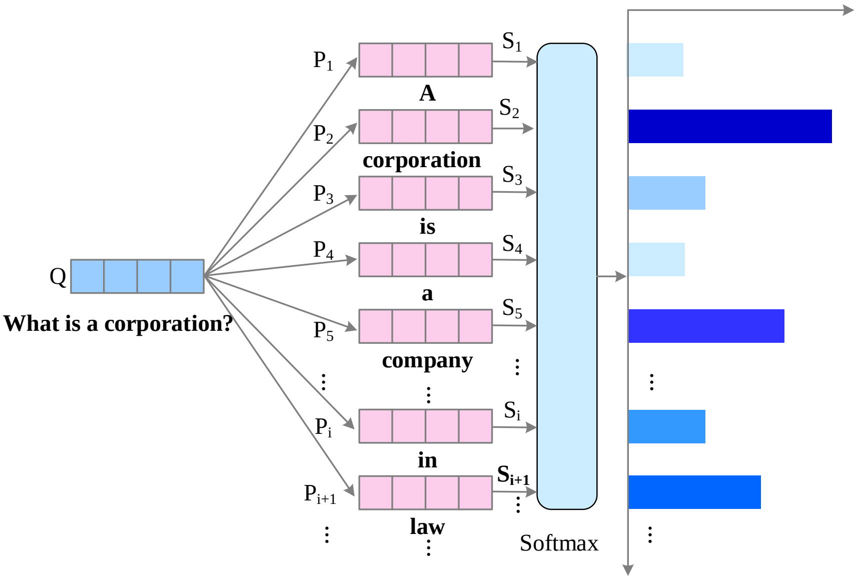


Figure 11. Using unidirectional attention to mine correlation between the context and question.

The choice of function $f(\cdot)$ differs in different models.

In the Attentive Reader proposed by Hermann et al. [5], a tanh layer is used to compute the relevance between the context and the question as follows:

$$S_i = \tanh(W_P P_i + W_Q Q), \tag{5}$$

where W_P and W_Q are trainable parameters.

Following the work of Hermann et al., Chen et al. [12] substitute a bilinear term for the tanh function as Equation (6):

$$S_i = Q^T W_s P_i, \tag{6}$$

which makes the model simpler and more effective than the Attentive Reader.

The unidirectional attention mechanism can highlight the most important context words to answer the question. However, this method fails to pay attention to question words that are also pivotal for answer prediction. Hence, unidirectional attention flow is insufficient for extracting mutual information between the context and the query.

(2) Bidirectional Attention

Seeing the limitations of the unidirectional attention mechanism, some researchers introduced bidirectional attention flows, which not only computes query-to-context attention but also the reverse, context-to-query attention. This method, which takes a mutual look from both directions, can benefit from the interaction between context and query and provide complementary information.

Figure 12 presents the process of computing bidirectional attention. First, the pair-wise matching matrix $M(i,j)$ is obtained by computing the matching scores between each context semantic embedding P_i and question semantic embedding Q_j (by word-level encoding introduced in Section 3.2.2). Then the outputs of the column-wise SoftMax function can be regarded as query-to-context attention weight α while context-to-query attention β is calculated by the row-wise SoftMax function.

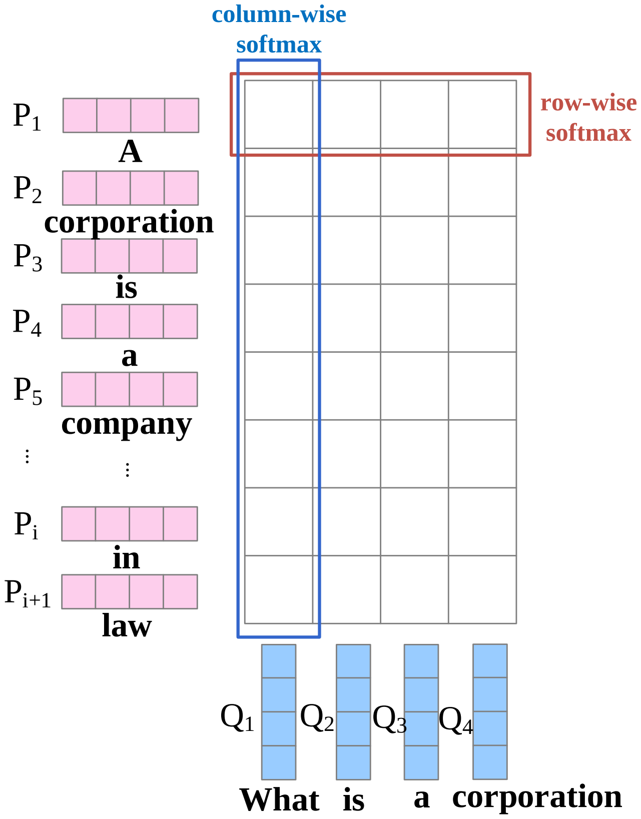


Figure 12. Using bidirectional attention to mine correlation between the context and question.

The attention-over-attention reader (AoA Reader) model, the dynamic coattention network (DCN) and the bidirectional attention flow (Bi-DAF) network are representative MRC models with bidirectional attention.

In the AoA Reader, Cui et al. [18] compute the dot product between each context embedding and query embedding to obtain a similarity matching matrix $M(i,j)$. The query-to-context and context-to-query attention are calculated as presented in Figure 12. To combine these two attentions, different from previous work in CAS Reader [53] using naive heuristics such as sum and average over the query-to-context attention, Cui et al. introduce attended attention, computed by the dot product of α and the average result of β, which is later used to predict the answer.

To attend to the question and the document simultaneously, Xiong et al. [17] fuse the two directional attentions as follows:

$$C = \alpha[Q;\beta P], \tag{7}$$

where C can be regarded as the coattention representations that contains attention information of both context and question. Based on DCN, Xiong et al. [22] introduce an extension, DCN+, using residual connections to merge coattention outputs to encode richer information for the input sequences. Compared to the work on AoA Reader, Xiong et al. further calculate context representations with two types of directional attentive information, rather than directly using attention weights for answer prediction, which can better extract the correlation between context and question.

In contrast to the AoA Reader and DCN, which directly summarize the output of two directional attention flows, Seo et al. [16] let attentive vectors flow into another RNN layer to encode query-aware context representations, which can reduce information loss caused by early summarization. More specifically, after obtaining query-to-context attention weight α and context-to-query attention weight β, Seo et al. compute attended context vector $\widetilde{P}$ and attended query vector $\widetilde{Q}$ as follows:

$$\begin{aligned} \widetilde{P} &= \sum_i \alpha P_i, \\ \widetilde{Q} &= \sum_j \beta Q_j. \end{aligned} \tag{8}$$

Then the context embeddings and attention vectors are combined by a simple concatenation:

$$G = [P; \widetilde{Q}; P \circ \widetilde{Q}; P \circ \widetilde{P}], \tag{9}$$

where $\circ$ is element-wise multiplication and G can be regarded as query-aware context representations, which are later fed to bidirectional LSTM to be further encoded.

To sum up, MRC systems at the early stage usually used unidirectional attention mechanism, especially query-to-context attention, to highlight the part of the context that was more important to answer the question. However, query-to-context attention is not sufficient to extract the mutual information between the context and the query. Later, bidirectional attention was widely applied to overcome the shortcomings of unidirectional attention, which benefited from context-query correlation and output attentive representations with a fusion of context and question information.

(3) One-Hop Interaction

One-hop interaction is a shallow architecture, where the interaction between the context and the question is computed only once. Earlier context-query interaction was such a one-hop architecture in many MRC systems; for example, the Attentive Reader [5], the attention sum (AS) Reader [13], the AoA Reader [18] and so on. Although this method can do well in tackling simple cloze tests, when the question requires reasoning over multiple sentences in the context, it is hard for this one-hop interaction approach to predict the correct answer.

(4) Multi-Hop Interaction

In contrast to one-hop interaction, multi-hop interaction is much more complex; it tries to mimic the rereading phenomenon of humans by computing the interaction between the context and the question more than once. In the process of the interaction, whether information of the previous state, i.e., the memory of the already read context and question, can be efficiently stored or not directly affects the performance of the next interaction.

There are mainly three methods to perform multi-hop interaction:

The first method calculates the similarity between the context and the question based on previous attentive representations of context. In the Impatient Reader model proposed by Hermann et al. [5], query-aware context representations are dynamically updated by this method as each query token is read. This simulates the process of humans rereading a given context with the question information.

The second method introduces external memory slots to store previous memories. The representative model using this method is memory networks, proposed by Weston et al. [54], which can explicitly store long-term memories and has easy access to reading memories. With such a mechanism, MRC models can have a deeper understanding of the context and the question by multiple turns of interaction. After being given the context as input, the memory mechanism stores information of the context in memory slots and then updates them dynamically. The process of answering involves finding the memory most relevant to the question and turning it into an answer representation as required. Although this method can overcome the shortcoming of insufficient memory, it is hard to train the network via back-propagation. To address this problem, Sukhbaatar et al. [11] introduce an end-to-end version of memory networks, in which explicit memory storage is embedded with continuous representations and the process of reading and updating memories is modeled by neural networks. This extension of memory networks can reduce supervision during training and is applicable to more tasks.

The characteristic of updating memories with multiple hops in memory networks makes this method popular in MRC systems. Pan et al. [20] propose the MEMEN model, which stores question-aware context representations, context-aware question representations, and candidate answer representations in memory slots and updates them dynamically. Similarly, Yu et al. [55] use external memory slots to store question-aware context representations and update memories with bidirectional GRUs.

The third method takes advantage of the recurrence feature of RNNs, using hidden states to store previous interaction information. Wang and Jiang [15] perform multiple interaction by using match-LSTM architecture recurrently. This model is originally proposed for textual entailment; when introduced to MRC, it can simulate the process of reading passages with question information. First, Wang and Jiang use the standard attention mechanism to obtain the attentive weight of each context token for the question. After calculating the dot product of question tokens and attentive weights, the model concatenates it with the context token and feeds it to a match-LSTM to get query-aware context representations. Similarly, this process is done in the reverse direction to fully encode contextual information. Finally, the outputs of match-LSTM in two directions are concatenated and fed to the answer prediction module. In addition, R-NET [21] and iterative alternating reader (IA Reader) [14] also use RNNs to update query-aware context representations to perform multi-hop interaction.

Some early work treated each context and query word equally when mining their correlations. However, the most important part should be given more attention for efficient context-query interaction. The gate mechanism, which can control the amount of mutual information between the context and the question, is a key component in multi-hop interaction.

For the gated-attention reader (GA Reader), Dhingra et al. [19] use the gate mechanism to decide how question information affects the focus on context words when updating context representations. The gate attention mechanism is performed by an element-wise multiplication between query embeddings and intermediate representations of context more than once.

In contrast to the GA Reader, context and question representations are updated in the iterative alternating attention mechanism [14]. Question representations are updated with previous search states, while context representations are refined with previous reasoning information as well as currently updated queries. Then the gate mechanism, which is performed by a feed-forward network, is applied to determine the degree of matching between the context and the query. This mechanism is capable of extracting evidence from the context and the question alternately.

Previous models ignored the fact that context words have different importance when answering particular questions. To address this problem, Wang et al. [21] introduce the gate mechanism to filter out insignificant parts in the context and emphasize the ones most relevant to the question in their R-NET model. This model can be regarded as a variant of the attention-based recurrent networks. Compared to match-LSTM [15], it introduces an additional gate mechanism based on current context representations and context-aware question representations. Moreover, as RNN-based models cannot deal well with long documents because of insufficient memory, Wang et al. add self-attention to the context itself. This mechanism can dynamically refine context representations based on mutual information from the whole context and the question.

In conclusion, one-hop interaction may fail to comprehensively understand mutual question-context information. By contrast, multiple-hop interaction with the memory of previous contexts and questions is capable of deeply extracting correlations and aggregating evidence for answer prediction.

3.2.4. Answer Prediction

This module is always the last of the MRC systems to give answers to questions according to the original context. The implementation of answer prediction is highly task-specific. Just as MRC tasks are categorized into cloze tests, multiple choice, span extraction, and free answering, there are four answer prediction methods: word predictor, option selector, span extractor, and answer generator. In this part, we will give an illustration in detail.

(1) Word Predictor

Cloze tests require filling in blanks with missing words or entities; one is asked to find out a word or an entity from the given context as the answer. In early work, such as the Attentive Reader [5], the combination of the query-aware context and the question is reflected in the vocabulary space to search for the correct answer word. Chen et al. [12] directly use query-aware context representations to match the candidate answer, simplifying the prediction process and improving the performance.

This method employs attentive context representations to select the correct answer word, but it cannot ensure that the answer is in the context, which does not satisfy the requirements of cloze tests. A related example is shown below (Kadle et al. [13]).

Context: *A UFO was observed above our city in January and again in March.*
Question: *An observer has spotted a UFO in _____.*

In the situation where both *January* and *March* can be the correct answer, methods used by Hermann et al. [5] and Chen et al. [12], which reflect attentive context representations into the whole vocabulary space, would give an answer similar to these two words, maybe *February* because of features of distributed word representation pre-trained by Word2Vec.

To overcome the problem that the predicted answer may not be in the context, Kadlec et al. [13] propose the attention sum reader (AS Reader) model, inspired by pointer networks. Pointer networks, introduced by Vinyals et al. [56], are adapted to tasks whose outputs can only be selected from inputs at first and can satisfy the requirements of cloze tests. In the AS Reader, Kadlec et al. do not compute the attentive representations; instead, they directly use attention weights to predict the answer. The attention results of the same word are added together, and the one with the maximum value is selected as the answer. This method is simple but quite efficient for cloze tests.

(2) Option Selector

To tackle the multiple-choice task, the model should select the correct answer from candidate options. The common way is to measure the similarity between attentive context representations and candidate answer representations and the most similar candidate is chosen as the correct answer.

Chaturvedi et al. [28] use CNNs to encode question-option tuples and relevant context sentences. Then they measure the correlation between them by cosine similarity, and the most relevant option is selected as the answer. Zhu et al. [29] introduce the information of options to contribute to extracting the interaction between the context and the question. In the answer prediction module, they use a bilinear function to score each option according to the attentive information; the one with the highest score is the predicted answer. In the convolutional spatial attention model, Chen et al. [27] calculate similarities among question-aware candidate representations, context-aware representations, and self-attended question representations with dot product to fully extract correlations among context, question, and options. The various similarities are concatenated and then fed to CNNs with different kernel sizes. The outputs of the CNNs are regarded as feature vectors and fed to fully connected layers to calculate a score for each candidate. Finally, the correct answer is the one with highest score.

(3) Span Extractor

The span extraction task can be regarded as an extension of cloze tests, which requires extracting a subsequence from the context rather than a single word. As word predictor methods used in models for cloze tests can only extract one context token, they cannot be directly applied to the span extraction task. Also inspired by pointer networks [56], Wang and Jiang [15] propose two models, the sequence model and the boundary model to overcome the shortcomings of word prediction approaches. Outputs of the sequence model are positions where answer tokens appear in the original context. The process of answer prediction is similar to the decoding of sequence-to-sequence models, selecting tokens with the highest probability successively until it stops generating answer tokens. Answers obtained by such methods are treated as a sequence of tokens from the input context, which might not be a consecutive span and cannot be ensured to be a subsequence of original context. The boundary model can handle the problem, which only predicts the start and end positions of the answer. The boundary model is much simpler and shows better performance on SQuAD. It has been widely used in other MRC models as a preferred alternative for subsequence extraction.

Considering that there is more than one plausible answer span in the original context, but the boundary model might extract incorrect answers with local maxima, Xiong et al. [17] proposed a dynamic pointing decoder to select an answer span by multiple iterations. This method uses LSTM to estimate the start and end positions based on representations corresponding to the last state answer prediction. To compute start and end score of context tokens, Xiong et al. propose highway maxout networks (HMN) with maxout networks [57] and highway networks [58], which require different models according to various question types and context topics.

(4) Answer Generator

With the appearance of free answering tasks, answers are no longer limited to a subsequence of the original context; instead, they need to be synthesized from the context and the question. Specifically, the expression of answers may differ from the evidence snippet in the given context, or answers may be from multiple pieces of evidence in different passages. Answer forms of the free answering task have the fewest limits, but this task has high requirements for the answer prediction module. To deal with the challenge, some generation approaches have been introduced to generate flexible answers.

S-NET, proposed by Tan et al. [23], introduces the answer generation module to satisfy the requirement of free answering tasks, whose answers are not limited to the original context. It follows the “extraction and then synthesis” process. The extraction module is a variant of R-NET, [21] while the generation module is a sequence-to-sequence architecture. More specifically, for the encoder, bidirectional GRUs are used to produce context and question representations. In particular, start and end positions of evidence snippets predicted by the span extraction module are added to context representations as additional features. In terms of the decoder, the state of the GRUs is updated by the previous context word representations and attentive intermediate information. After applying the SoftMax function, the output of the decoder is the synthetic answer.

The generation module successfully makes up for the deficiency of the extraction module and generates more flexible answers. However, answers generated by existing approaches may suffer from syntax errors and faulty logic problems. Hence, generation and extraction methods are usually used together to provide complementary information. For example, in S-NET, the extraction module first labels the approximate boundary of the answer span, while the generation module generates answers not limited to the original context based on that. Generation approaches are not very common in existing MRC systems, as extraction methods already perform well enough in most cases.

3.3. Additional Tricks

While some typical deep-learning methods have been introduced here, there are some additional tricks, such as reinforcement learning, answer ranker, and sentence selector, that cannot be included in the general MRC architecture. However, these tricks also contribute to performance improvement, so we will give brief descriptions of them in the following sections.

3.3.1. Reinforcement Learning

As can be seen from the above introduction, most MRC models only apply maximum-likelihood estimation in the training process. However, there is a disconnect between optimization objectives and evaluation metrics. As a result, candidate answers that exactly match the ground truth or have word overlap with the ground truth but are not located at the labeled positions would be ignored by such models. In addition, when the answer span is too long or has fuzzy boundaries, models would also fail to extract the correct answer. MRC evaluation metrics, such as exact match (EM) and F1, are not differentiable, so some researchers introduce reinforcement learning to the training process. Xiong et al. [22] and Hu et al. [26] use F1 score as the reward function and treat maximum-likelihood estimation and reinforcement learning as a multi-task learning problem. This method can consider both textual similarity and position information.

Reinforcement learning can also be used to determine whether to stop the interaction process. Multi-hop interaction methods, introduced above, have a pre-defined number of hops in the interaction. However, when people are answering a question, they stop reading if there is adequate evidence for the answer. The termination state is highly related to the complexity of the given context and question. Motivated by stopping interaction dynamically according to the context and question, Shen et al. [59] introduce a termination state to their ReasonNets. If the value of this state equals 1, the model stops the interaction and feeds the evidence to the answer prediction module to give the answer; otherwise ReasonNets continues the interaction by computing the similarity between the intermediate state and the input context and query. As the termination state is discrete and back-propagation cannot be used directly while training, reinforcement learning is applied to train the model by maximizing the instance-dependent expected reward.

In a word, reinforcement learning can be regarded as an improved approach in MRC systems that is capable of not only reducing the gap between optimization objectives and evaluation metrics but also determining whether to stop reasoning dynamically. With reinforcement learning, the model can be trained and refine better answers even if some states are discrete.

3.3.2. Answer Ranker

To verify whether the predicted answer is correct or not, some researchers introduce an answer ranker module. The common process of ranker is that it extracts some candidate answers, and the one with the highest score is the correct answer.

EpiReader [60] combines pointer methods with the ranker. Trischler et al. extract answer candidates using an approach similar to the AS Reader [13], selecting some answer spans with the highest attention sum score. Then EpiReader feeds those candidates to the reasoner component, which inserts them to the question sequence at placeholder location and computes their probability to be the correct answers. The one with the highest probability is selected as the correct answer.

To extract candidates with variable lengths, Yu et al. [61] propose two approaches. In the first one, they capture the part-of-speech (POS) patterns of answers in the training set and choose subsequences in the given passage that could match such patterns as candidates. The other approach enumerates all possible answer span within a fixed length from the context. After obtaining answer candidates, Yu et al. compute their similarity with question representations and choose the most similar one as the answer.

With the ranker module, the accuracy of answer prediction can be improved somewhat. These methods have also inspired some researchers to detect unanswerable questions.

3.3.3. Sentence Selector

In practice, if the MRC model is given a long document, it takes a lot of times to understand the full context to answer questions. However, finding the sentences that are most relevant to the questions in advance is a possible way to accelerate the training process. With this motivation, Min et al. [62] propose a sentence selector to find the minimal set of sentences needed to answer a question. The architecture of the sentence selector is sequence-to-sequence, which contains an encoder to compute sentence and question encodings and a decoder to calculate a score for each sentence by measuring the similarity between the sentence and the question. If the score is higher than a pre-defined threshold, the sentence is selected to be fed to the MRC systems. In this way, the number of selected sentences is dynamic according to different questions.

An MRC system with a sentence selector is capable of reducing training and inference time with equivalent or better performance compared to a system without a sentence selector.

4. Datasets and Evaluation Metrics

The release of large-scale MRC datasets makes it possible to train deep neural models while evaluation metrics can evaluate the performance of MRC models, both of which play significant role in the MRC field. In this section, we describe representative datasets according to different tasks, then introduce evaluation metrics.

4.1. Datasets

Datasets are among the driving factors accelerating the development of the MRC field, some of which, such as CNN & Daily Mail [5], SQuAD [6] and MS MARCO [7], can be regarded as milestones of MRC, spurring the emergence of up-to-date techniques. In this part, we introduce several representative datasets of each MRC task, highlighting how to construct large-scale datasets according to task requirements, and how to reduce lexical overlap between questions and context.

4.1.1. Cloze Tests Datasets

- CNN & Daily Mail

This dataset, built by Hermann et al. [5], is one of the most representative cloze-style MRC datasets. CNN & Daily Mail, consisting of 93,000 articles from the CNN and 220,000 articles from the Daily Mail, is indeed large-scale and makes it possible to use deep-learning approaches in MRC. Considering that bullet points are abstractive and have little sentence overlap with documents, Hermann et al. replace one entity at a time with a placeholder in bullet points and evaluate the machine reading system by asking the machine to read the documents and then predict which entity the placeholder in the bullet points referred to. As questions are not posed directly from documents, this task is challenging, and some information extraction methods fail to deal with it. This methodology of creating MRC datasets has enlightened a lot of other research [63–65]. To avoid the situation where questions can be answered by knowledge outside of the documents, all entities in documents are anonymized by random markers.

- CBT

Hill et al. [66] design a cloze-style MRC dataset, the Children's Book Test (CBT), from another perspective. They collect 108 children's books and form each sample with 21 consecutive sentences from chapters in those books. To generate questions, a word from the 21st sentence is removed, and the other 20 sentences act as context. Nine incorrect words, whose type is the same as the answer, are selected at random from the context as candidate answers. There are some differences between the

CNN & Daily Mail and CBT dataset. First, unlike the CNN & Daily Mail, entities in the CBT dataset are not anonymized, so models can use background knowledge from wider contexts. Second, missing items in the CNN & Daily Mail are limited to named entities, but in the CBT there are four distinct types: named entities, nouns, verbs, and prepositions. Third, the CBT provides candidate answers, which simplifies the task in a way. Overall, with the appearance of the CBT, context, which plays a significant role in human comprehension, has gained much more attention. Considering that more data can significantly improve the performance of neural network models, Bajgar et al. [67] introduce the BookTest, which enlarges the CBT dataset 60 times and enables training of larger models.

- LAMBADA

To account for the meaning of the wider context, Paperno et al. [68] propose the language modeling broadened to account for discourse aspects (LAMBADA) dataset. Similar to the CBT, LAMBADA also uses books as the source, and the task is word prediction. However, the word that needs to be predicted in LAMBADA is the last word in the target sentence, while in CBT any word in the target sentence may be targeted. Moreover, Paperno et al. find that some samples in the CBT can be guessed with the target sentence alone rather than in a wider context. To overcome this shortcoming, there is a constraint in LAMBADA that makes it difficult to predict the target word correctly only with the target sentence. That is to say, compared with CBT, LAMBADA requires more understanding of the wider context.

- Who-did-What

To better evaluate the understanding of natural language, researchers try to avoid sentence overlap between questions and documents when constructing MRC datasets. Onishi et al. [64] provide new insight into how to reduce the syntactic similarity. In the "Who-did-What" dataset, each sample is formed from two independent articles; one serves as the context and questions are generated from the other. This approach can be used by other corpora, in which articles do not have summary points, unlike CNN & Daily Mail. There is another feature of Who-did-What, as shown in the name; the dataset only pays attention to the personal name entity, which may be its limitation.

- CLOTH

In contrast to the above automatically generated datasets, cloze test by teachers (CLOTH) [69] is human-created, collected from English exams for Chinese students. Questions in CLOTH are well-designed by middle-school and high-school teachers to examine students' language proficiency, including vocabulary, reasoning, and grammar. There are fewer purposeless or trivial questions in CLOTH, so that it requires a deep understanding of language.

- CliCR

To address the problem of scarce datasets for specific domains, Suster et al.[63] build a large-scale cloze-style dataset based on clinical case reports for healthcare and medicine. Similar to the CNN & Daily Mail, summary points of case reports are used to create queries by blanking out medical entities. CliCR promotes MRC to practical applications such as clinical decision-making.

4.1.2. Multiple-Choice Datasets

- MCTest

MCTest, proposed by Richardson et al. [70], is a multiple-choice MRC dataset at an early stage. It consists of 500 fictional stories, and for each story there are four questions with four candidate answers. Choosing fictional stories avoids introducing external knowledge, and questions can be answered according to the stories themselves. This idea of using a story-based corpus has inspired other datasets, such as CBT [66] and LAMBADA [68]. Although the appearance of MCTest has encouraged research on MRC, its size is too small, so it is not suitable for some data-hungry techniques.

- RACE

Like the CLOTH dataset [69], RACE [10] is also collected from English exams for middle-school and high-school Chinese students. This corpus allows a greater variety of types of passages. In contrast to one fixed style for the whole dataset, such as news for CNN & Daily Mail [5] and NewsQA [71], and fictional stories for CBT [66] and MCTest [70], almost all kinds of passages can be found in RACE. As a multiple-choice task, RACE asks for more reasoning, because questions and answers are human-generated and simple methods based on information retrieval or word co-occurrence may not perform well. In addition, compared to MCTest [70], RACE, which contains about 28,000 passages and 100,000 questions, is large-scale and supports the training of deep-learning models. All the aforementioned features illustrate that RACE is well-designed and full of challenges.

4.1.3. Span Extraction Datasets

- SQuAD

The Stanford Question-Answering Dataset (SQuAD), proposed by Rajpurkar et al.[6] of Stanford University, can be regarded as a milestone of MRC. With the release of SQuAD dataset, an MRC competition based on it has drawn the attention of both academia and industry, which in turn has stimulated the development of various advanced MRC techniques. Collecting 536 articles from Wikipedia, Rajpurkar et al. ask crowd-workers to pose more than 100,000 questions and select a span of arbitrary length from the given article to answer the question. SQuAD is not only large but also high quality. In contrast to prior datasets, SQuAD defines a new kind of MRC task, which does not provide answer choices, but needs a span of text as the answer rather than a word or an entity.

- NewsQA

NewsQA [71] is another span extraction dataset similar to SQuAD, in which questions are also human-generated and answers are spans of text from corresponding articles. The obvious difference between NewsQA and SQuAD is the source of articles. In NewsQA, articles are collected from CNN, while the SQuAD is based on Wikipedia. It is worth mentioning that some questions in NewsQA have no answer according to the given context. The addition of unanswerable questions makes it closer to reality and inspires Rajpurkar et al. [72] to update SQuAD to version 2.0. In terms of unanswerable questions, we give a detailed introduction in Section 5.2.

- TriviaQA

The construction process of TriviaQA [73] distinguishes it from previous datasets. In prior work, crowd-workers are given articles and pose questions closely related to those articles. However, this process results in the dependence of questions and evidence to answer them. Furthermore, in human understanding, people often ask a question and then find useful resources to answer it. To overcome this shortcoming, Joshi et al. gather question-answer pairs from trivia and quiz-league websites. Then they search for evidence to answer questions from webpages and Wikipedia. Finally, they build more than 650,00 question-answer-evidence triples for the MRC task. This novel construction process makes TriviaQA a challenging testbed with considerable syntactic variability between questions and contexts.

- DuoRC

Saha et al. [65] also try to reduce lexical overlap between questions and contexts in DuoRC. As in Who-did-What [64], questions and answers in DuoRC are created from two different versions of documents corresponding to the same movie, one from Wikipedia and one from IMDb. Asking questions and labeling answers are done by different group of crowd-workers. The distinction between the two versions of movie plots requires more understanding and reasoning. Moreover, there are unanswerable questions in DuoRC.

4.1.4. Free Answering Datasets

- bAbI

bAbI, proposed by Weston et al. [74], is a well-known synthetic MRC dataset. It consists of 20 tasks, generated with a simulation of a classic text adventure game. Each task is independent from the others and tests one aspect of text understanding, such as recognizing two or three argument relations and using basic deduction and induction. Weston et al. think that dealing with all these tasks is a prerequisite to full language understanding. Answers are limited to a single word or a list of words and may not be directly found from the original context. The release of the bAbI dataset promotes the development of several promising algorithms, but as all data in bAbI is synthetic, it is a little far from the real world.

- MS MARCO

MS MARCO [7] can be viewed as another milestone of MRC after SQuAD [6]. To overcome the weaknesses of previous datasets, it has four predominant features. First, all questions are collected from real user queries. Second, for each question, 10 related documents are searched with the Bing search engine to serve as the context. Third, labeled answers to those questions are generated by humans so that they are not restricted to spans of the context and more reasoning and summarization are required. Fourth, there are multiple answers to each question and sometimes they even conflict, which makes it more challenging for the machine to select the correct answer. MS MARCO makes the MRC dataset closer to the real world.

- SearchQA

The work of SearchQA [75] is just like TriviaQA [73]; both follow the general pipeline of question answering. To construct SearchQA, Dunn et al. collect question-answer pairs from the J!Archive and then search for snippets related to questions from Google. However, the major difference between SearchQA and TriviaQA is that in TriviaQA there is one document with evidence for each question-answer pair, while in SearchQA each pair has 49.6 related snippets on average.

- NarrativeQA

Seeing the limitation in most previous datasets of needing evidence to answer questions from single sentences of original context, Kovcisky et al. [76] design the NarrativeQA. Based on book stories and movie scripts, they search related summaries from Wikipedia and ask co-workers to generate question-answer pairs according to those summaries. What makes NarrativeQA special is that answering questions requires understanding the whole narrative, rather than superficial matching information.

- DuReader

Similar to MS MARCO [7], DuReader, released by He et al. [77], is another large-scale MRC dataset from real-world application. Questions and documents in DuReader are collected from Baidu Search (search engine) and Baidu Zhidao (question-answering community). Answers are human-generated instead of spans of original context. What makes DuReader different is that it provides new question types such as yes/no and opinion. Compared to factoid questions, these questions sometimes require summaries over multiple parts of documents, which opens an opportunity for the research community.

4.2. Evaluation Metrics

For different MRC tasks, there are various evaluation metrics. To evaluate cloze tests and multiple-choice tasks, the most common metric is accuracy. In terms of span extraction, exact match (EM), a variant of accuracy, and F1 score are computed to measure model performance. Considering that answers for free answering tasks are not limited to the original context, ROUGE-L and BLEU are widely used. In the following part, we will give detailed descriptions of these evaluation metrics.

- Accuracy

Accuracy with respect to ground-truth answers is usually applied to evaluate cloze tests and multiple-choice tasks. When given a question set $Q = \{Q_1, Q_2, \cdots Q_m\}$ with m questions, if the model correctly predicts answers for n questions, then accuracy is calculated as follows:

$$\text{Accuracy} = \frac{n}{m}. \tag{10}$$

Exact match is a variant of accuracy that evaluates whether a predicted answer span matches the ground-truth sequence exactly or not. If the predicted answer is equal to the gold answer, the EM value will be 1, otherwise 0. It can also be calculated by the above equation.

- F1 Score

F1 score is a common metric in classification tasks. In terms of MRC, both candidate and reference answers are treated as bags of tokens and true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are denoted as shown in Table 8.

Table 8. The definition of true positive (TP), true negative (TN), false positive (FP), false negative (FN).

	Tokens in Reference	**Tokens Not in Reference**
tokens in candidate	TP	FP
tokens not in candidate	FN	TN

Then precision and recall are computed as below:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{12}$$

F1 score, also known as balanced F score, is the harmonic average of precision and recall:

$$\text{F1} = \frac{2 \times P \times R}{P + R}, \tag{13}$$

where P denotes precision while R is recall.

Compared to EM, this metric loosely measures the average overlap between the prediction and the ground-truth answer.

- ROUGE-L

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an evaluation metric initially developed for automatic summarization, proposed by Lin and Chin-Yew [78]. It evaluates the quality of a summary by counting the amount of overlap between model-generated and ground-truth summaries. There are various ROUGE measures, such as ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S, for different evaluation requirements, among which ROUGE-L is widely used in MRC tasks with the use of free answering. Unlike other metrics, such as EM or accuracy, ROUGE-L is more flexible, mainly measuring similarities between gold answers and predicted answers. The "L" in ROUGE-L denotes longest common subsequence (LCS) and ROUGE-L can be computed as follows:

$$R_{\text{lcs}} = \frac{LCS(X,Y)}{m}, \tag{14}$$

$$P_{\text{lcs}} = \frac{LCS(X,Y)}{n}, \tag{15}$$

$$F_{\text{lcs}} = \frac{(1+\beta)^2 R_{\text{lcs}} P_{\text{lcs}}}{R_{\text{lcs}} + \beta^2 P_{\text{lcs}}}, \tag{16}$$

where X is ground-truth answer with m tokens, Y is model-generated answer with n tokens, $\text{LCS}(X,Y)$ denotes the length of the longest common subsequence of X and Y, and β is a parameter to control the importance of precision P_{lcs} and recall R_{lcs}.

Using ROUGE-L to evaluate the performance of an MRC model does not require predicted answers to be consecutive subsequences of ground truth, whereas more token overlap contributes to higher ROUGE-L scores. However, the length of candidate answers influences the value of ROUGE-L.

- BLEU

BLEU (Bilingual Evaluation Understudy), proposed by Papineni et al. [79], is widely used to evaluate translation performance. When adapted to MRC tasks, BLEU score measures the similarity between predicted answers and ground truth. The cornerstone of this metric is precision measure, which is calculated as follows:

$$P_n(C,R) = \frac{\sum_i \sum_k \min(h_k(c_i), \max(h_k(r_i)))}{\sum_i \sum_k h_k(c_i)}, \tag{17}$$

where $h_k(c_i)$ counts the number of k-th n-gram appearing in candidate answer c_i; in a similar way, $h_k(r_i)$ denotes the occurrence number of that n-gram in gold answer r_i.

As the value of $P_n(C,R)$ is higher when the answer span is shorter, such precision cannot measure the similarity well by itself. The penalty factor, BP, is introduced to alleviate that, which is computed as:

$$\text{BP} = \begin{cases} 1, l_c > l_r \\ e^{1-\frac{l_r}{l_c}}, l_c \leq l_r. \end{cases} \tag{18}$$

Finally, the BLEU score is computed as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right), \tag{19}$$

where N means n-grams up to length N and w_n equals $1/N$. The BLEU score is the weighted average of each n-gram and the maximum of N is 4, namely BLEU-4.

BLEU score can not only evaluate the similarity between candidate answers and ground-truth answers but also test the readability of candidates.

5. New Trends

Since about 2017, the research community has started to focus on developing neural models for some more complicated problems or tasks related to MRC, such as how to leverage knowledge bases and how to deal with unanswerable questions, with the belief that they are more significant for practical applications and that deep-learning techniques can address them more effectively, though some of them may have been involved in traditional expert or QA systems. We regard them as new trends within the context of neural MRC and discuss them in the following sections.

5.1. Knowledge-Based Machine Reading Comprehension

MRC requires answering questions with knowledge implicit in the given context. Datasets such as the MCTest choose passages from specific corpora (fiction stories, children's books, etc.) to avoid introducing external knowledge. However, human-generated questions are usually too simple when compared to questions in real-world application. In the process of human reading comprehension, we may use common sense when we cannot answer a question simply by knowing about the context. External knowledge is so significant that is believed to be the biggest gap between MRC and human reading comprehension. As a result, interest in introducing world knowledge to MRC has surged in the research community, and knowledge-based machine reading comprehension (KBMRC) has come into being. KBMRC differs from MRC mainly in terms of inputs. In MRC, the inputs are sequences of context and questions. However, besides that, additional related knowledge extracted from knowledge bases is necessary in KBMRC.

KBMRC can be regarded as augmented MRC with external knowledge K and it can be formulated as shown in Table 9:

Table 9. Definition of knowledge-based machine reading comprehension.

Knowledge-Based Machine Reading Comprehension
Given the context C, question Q and external knowledge K, the task requires predicting the correct answer A by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C, Q, K)$.

There are some KBMRC datasets in which world knowledge is needed to answer some questions. MCScripts [80] is a dataset about human daily activities, such as eating in a restaurant and taking a bus, where answering some questions requires common sense knowledge beyond the given context. As shown in Table 10, the answer to *Why are trees important?* cannot be found in the given context. However, it is common sense knowledge to us that trees are important because they *create oxygen* and absorb carbon dioxide via photosynthesis not because *they are green*.

Table 10. Some Examples in KBMRC

MCScripts	
Context:	Before you plant a tree, you must contact the utility company. They will come to your property and mark out utility lines. Without doing this, you may dig down and hit a line, which can be lethal! Once you know where to dig, select what type of tree you want. Take things into consideration such as how much sun it gets, what zone you are in, and how quickly you want it to grow. Dig a hole large enough for the tree and roots. Place the tree in the hole and then fill the hole back up with dirt$\cdots$
Question:	Why are trees important?
Candidate Answers:	A. create O_2 B. because they are green

The key challenges in KBMRC are as follows:

- Relevant External Knowledge Retrieval

There are various kinds of knowledge stored in knowledge bases, and entities may be misleading sometimes because of polysemy, e.g., “apple” can refer to a fruit or a corporation. Extracting knowledge closely related to the context and the question determines the performance of knowledge-based answer prediction.

- External Knowledge Integration

In contrast to text in the context and questions, knowledge in an external knowledge base has its own unique structure. How to encode such knowledge and integrate it with representations of the context and questions remains an ongoing research challenge.

Some researchers have tried to address the above challenges in KBMRC. To make the model take advantage of the external knowledge, Long et al. [81] propose a new task, rare entity prediction, which requires predicting missing name entities and is similar to cloze tests. However, name entities removed from the context cannot be predicted correctly based only on the original context. This task provides additional entity descriptions extracted from knowledge bases such as Freebase as external knowledge to help with entity prediction. While incorporating external knowledge, Yang and Mitchell [82] consider the relevance between knowledge and context to avoid irrelevant external knowledge misleading the answer prediction. They design the attention mechanism with a sentinel to determine whether to incorporate external knowledge or not and which knowledge should be adopted. Mihaylov and Frank [83] and Sun et al. [84] use key-value memory networks [85] to determine relevant external knowledge. All possible related knowledge is first selected from a knowledge base and stored in memory slots as key-value pairs. Then keys are used to match with the query, while corresponding values are summed together according to different weights to generate relevant knowledge representations. Wang and Jiang [86] propose a data enrichment method with semantic relations in WordNet, a lexical database for English. For each word in the context and question, they try to find the positions of passage words that have a direct or indirect semantic relationship with it to. This position information is regarded as external knowledge and fed to the MRC model to assist answer prediction.

In conclusion, KBMRC breaks through the limitation that the scope of knowledge required to answer questions is restricted to the given context. Hence, this task, benefit from external world knowledge, can close the gap between machine comprehension and human understanding to some extent. However, the performance of KBMRC systems is highly related to the quality of the knowledge base. Disambiguation is required when extracting related external knowledge from automated or semi-automated knowledge bases, as entities with the same name or alias could mislead the models. Moreover, knowledge stored in knowledge bases is usually sparse. If related knowledge cannot be found directly, incorporating external knowledge calls for further inference.

5.2. Machine Reading Comprehension with Unanswerable Questions

There is a latent hypothesis behind MRC tasks that correct answers always exist in the given context. However, this does not conform with real-world application. The range of knowledge covered in a passage is limited; thus, some questions inevitably have no answers according to the given context. A mature MRC system should distinguish those unanswerable questions.

MRC with unanswerable questions considers questions that cannot be answered based on the given context in a process consisting of two subtasks: answerability detection and reading comprehension. The definition of this new task is shown in Table 11:

Table 11. Definition of machine reading comprehension with unanswerable questions.

Machine Reading Comprehension with Unanswerable Questions
Given the context C and question Q, the machine first determines whether Q can be answered or not based on the given context C. If the question is impossible to be answered, the model marks it as unanswerable and abstain from answering, otherwise predicts the correct answer A by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(C, Q)$.

SQuAD 2.0 [72] is a representative MRC dataset with unanswerable questions. Based on the previous version released in 2016, SQuAD 2.0 has more than 50,000 unanswerable questions created by crowd-workers. Those questions, impossible to answer based on the context alone, are challenging, as they are relevant to the given context and there are plausible answer spans matches the question requirements. To perform well on SQuAD 2.0, a model should not only gives correct answers to answerable questions, but also detect which questions have no answers. An example of an unanswerable question in SQuAD 2.0 is presented in Table 12. In the context, the keywords *1937 treaty* exist and *Bald Eagle Protection Act* is the name of the treaty in 1940, not 1937, which is very puzzling.

Table 12. Unanswerable question example in SQuAD 2.0

SQuAD 2.0	
Context:	. . . Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protec -tion Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.
Question:	What was the name of the 1937 treaty
Plausible Answer:	Bald Eagle Protection Act

With unanswerable questions, there are two other challenges in this new task, compared to MRC:

- Unanswerable Question Detection

The model should know what it does not know. After comprehending the question and reasoning through the passage, the MRC models should judge which questions are impossible to answer based on the given context and mark them as unanswerable.

- Plausible Answer Discrimination

To avoid the impact of fake answers, such as the example presented in Table 12, the MRC model must verify the predicted answers and tell plausible answers from correct ones.

For the above two challenges, methods applied to tackle the problems in MRC with unanswerable questions can be categorized into two sorts:

To indicate no-answer cases, one approach employs a shared-normalization operation between a no-answer score and an answer span score. Levy et al. [87] add an extra trainable bias to the confidence score of start and end position and apply SoftMax to the new score to obtain the probability distributions of no answer. If this probability is higher than that of the best span, it means the question is unanswerable, otherwise it outputs the answer span. In addition, they propose another method that sets a global confidence threshold; if the predicted answer confidence is below the threshold, the model labels the question as unanswerable. Although this approach can detect unanswerable questions, it cannot guarantee that predicted answers are correct. The other methods introduce a no-answer option by padding. Tan et al. [88] add a padding position for the original passage to determine whether the question is answerable. When the model predicts that position, it refuses to give an answer.

Researchers also pay attention to the legitimacy of answers and have introduced answer verification to discriminate plausible answers. For unanswerable question detection, Hu et al. [89]

propose two types of auxiliary loss, independent span loss to predict plausible answers regardless of the answerability of the question, and independent no-answer loss, which alleviates the conflict between plausible answer extraction and no-answer detection tasks. In terms of answer verification, they introduce three methods. The first, sequential architecture, treats the question, answer, context sentence containing candidate answers as a whole sequence, and inputs that to the fine-tuned transformer model to predict the no-answer probability. The second is interactive architecture, which calculates the correlation between question-and-answer sentence in the context to classify whether the question is answerable. The third method integrates the above two approaches by concatenating the outputs of two models as joint representations, and this hybrid architecture can yield better performance.

In contrast to the above pipeline structure, Sun et al. [90] use multi-task learning to jointly train answer prediction, no-answer detection, and answer validation. What distinguishes their work is a combined universal node encoding passage and question information, which is then integrated with question representations and answer position aware passage representations. After being passed through the linear classification layer, fused representations can be used to determine whether the questions are answerable.

As the Chinese saying goes, *To know what it is that you know, and to know what it is that you do not know, that is wisdom.* The detection of unanswerable questions requires deep understanding of text and requires more robust MRC models, making MRC much closer to real-world application.

5.3. Multi-Passage Machine Reading Comprehension

In MRC tasks, the relevant passages are pre-identified, which contradicts the question-answering process of humans. People usually ask a question first and then search for possibly related passages, where they find evidence to give the answer. To overcome this shortcoming, Chen et al. [45] extend MRC to machine reading at scale, more widely called multi-passage machine reading comprehension, which does not give one relevant passage for each question, unlike the traditional task. This extension can be applied to tackle open-domain question-answering tasks based on a large corpus of unstructured text. With its appearance, some multi-passage MRC task-specific datasets have been released, such as MS MARCO [7], TriviaQA [73], SearchQA [75], DuReader [77], and QUASAR [91].

In contrast to MRC, in a multi-passage MRC setting, the given context C is not a single passage, but a collection of documents $\mathbb{D}$. The definition of multi-passage MRC tasks changes as shown in Table 13:

Table 13. Definition of multi-passage machine reading comprehension.

Multi-Passage Machine Reading Comprehension
Given a collection of m documents $\mathbb{D} = \{D_1, D_2, \cdots, D_m\}$ and the question Q, the multi-passage MRC task requires giving the correct answer A to question Q according to documents $\mathbb{D}$ by learning the function $\mathcal{F}$ such that $A = \mathcal{F}(\mathbb{D}, Q)$.

Compared to MRC tasks, multi-passage MRC is far more challenging. For instance, although the DrQA model [45] achieves exact match accuracy of 69.5 on SQuAD, when applied to an open-domain setting (using the whole Wikipedia corpus to answer the question), its performance drops dramatically. The unique features of multi-passage MRC listed below are the main reasons for the degradation:

- Massive Document Corpus

This is the most prominent feature of multi-passage MRC, which makes it distinct from MRC, with one related passage. Under this circumstance, whether a model can retrieve the most relevant documents from the corpus quickly and correctly decides the final performance of reading comprehension.

- Noisy Document Retrieval

Multi-passage MRC can be regarded as a distantly supervised open-domain question-answering task that may suffer from noise issues. Sometimes the model may retrieve a noisy document that contains the correct answer span, but it is not related to the question. This noise will mislead the understanding of the context.

- No Answer

When the retrieval component does not perform well, there will be no answers in the document. If the answer extraction module ignores that, it outputs an answer even if it is incorrect, which will lead to performance degradation.

- Multiple Answers

In the open-domain setting, it is common to find multiple answers for a single question. For example, when asking *Who is the president of the United States*, both *Obama* and *Trump* are possible answers, but which is the correct answer requires reasoning based on the context.

- Evidence Aggregation

In terms of some complicated questions, snippets of evidence can appear in different parts of one document or even in different documents. To answer such questions correctly, a multi-passage MRC model needs to aggregate all the evidence. More documents mean more information, which contributes to more complete answers.

To address multi-passage MRC problems, one method follows the pipeline of "retrieve then read." More specifically, the retrieval component first returns several relevant documents, which are then proposed by the reader to give the answer. DrQA, introduced by Chen et al. [45], is a typical pipeline-based multi-passage MRC model. In the retrieval component, they use TF-IDF to select five relevant Wikipedia articles for each question in SQuAD to narrow the search space. For the reader module, they improve the model proposed in 2016 [12] with rich word representations and a pointer module to predict the begin and end positions of answer spans. To make scores of candidate spans throughout different passages comparable, Chen et al. use unnormalized exponential and argmax functions to choose the best answer. In this approach, retrieval and reading are performed separately, but errors made in the retrieval stage are easily propagated to the next reading component, which leads to performance degradation.

To alleviate error propagation caused by poor document retrieval, one way is to introduce the ranker component, and the other is to jointly train the retrieval and reading processes.

In terms of the ranker component, which re-ranks the documents retrieved by the search engine, Htut et al. [92] introduce two rankers, InferSent and Relation-Networks Rankers. The first uses a feed-forward network to measure the general semantic similarity between the context and question, while the second applies relation-networks to capture local interactions between context words and question words. Inspired by Learning to Rank research, Lee et al. [93] propose a Paragraph Ranker mechanism, which uses bidirectional LSTMs to compute representations of passages and questions and measures similarities between the passages and questions by dot product to score each passage.

For joint training, Reinforced Ranker-Reader (R^3), proposed by Wang et al. [94], is the representative model. In R^3, match-LSTM [15] is applied to compute the similarity between the question and each passage to obtain document representations, which are later fed to the ranker and the reader. In the ranker module, reinforcement learning is used to select the most relevant passage, while the function of the reader is to predict the answer span from this selected passage. These two tasks are trained jointly to mitigate error propagation caused by wrong document retrieval.

However, retrieval components in the above models have low efficiency. For example, DrQA [45] simply uses traditional IR approaches in the retrieval component, and R^3 [94] applies question-dependent passage representations to rank the passages. The computational complexity increases as the document corpus becomes larger. To accelerate the retrieval process, Das et al. [95]

propose a fast and efficient retrieval method that represents passages independent from questions and stores outputs offline. When given the question, the model computes a fast inner product to measure similarities between passages and questions. Then the top-ranked passages are fed to the reader to extract answers. Another unique characteristic of this work is the iterative interaction between the retriever and the reader. They introduce a gated recurrent unit to reformulate query representations to account for the state of the reader and the original query. The new query representations are then used to retrieve other relevant passages, which facilitates the reread process across the corpus.

In the multi-passage setting, there may be more than one possible answer, among which some are not correct answers to the question. Instead of selecting the first match span as the correct answer, Pang et al. [96] propose three heuristic methods. The RAND operation treats all answer spans equally and chooses one randomly, while the MAX operation chooses the one with maximum probability and can be used if there are noisy paragraphs. The SUM operation assumes that more than one span can be regarded as ground-truth and sums all span probabilities together. Similar to the MAX operation, Clark and Gardner [40] regard all labeled answer spans as correct and, inspired by attention sum reader [13], they use a summed objective function to choose the one with maximum probability to be the correct answer. In contrast, Lin et al. [97] introduce a fast paragraph selector to filter out passages with wrong answer labels before feeding them to the reader module. They use multi-layer perceptron or RNNs to obtain hidden representations of passages and questions, respectively. In addition, a self-attention operation is applied to the questions to illustrate their different importance. Then the similarity between passages and questions is calculated and the most similar ones are chosen as relevant passages to be fed to the reader module.

Wang et al. [98] see the significance of evidence aggregation in multi-passage MRC tasks. In their point of view, on the one hand, correct answers have more evidence across different passages. On the other hand, some questions require various aspects of evidence to answer. To make full use of multiple pieces of evidence, they propose strength-based and coverage-based re-rankers. In the first mechanism, the answer with the most occurrences among the candidates is chosen to be the correct one. The second re-ranker concatenates all passages that contain candidate answers as a new context and feeds that to the reader to obtain the answer which aggregates different aspects of evidence.

To sum up, compared to MRC tasks, multi-passage MRC is much closer to real-world application. With several documents given as resources, there is more evidence for answer prediction, thus even if the question is complicated, the model can give the answer fairly well. Related document retrieval is important in multi-passage MRC, and evidence aggregated from documents may be complementary or contradictory. Hence, free answering, in which answers are not limited to the subsequence in the original context, is common in multi-passage MRC tasks. Taking advantage of multiple documents and generating answers with the proper logic and clear semantics to answer questions still has a long way to go.

5.4. Conversational Machine Reading Comprehension

MRC requires answering a question based on the understanding of a given passage, with questions usually isolated from each other. However, the most natural way that people acquire knowledge is via a series of interrelated question-and-answer processes. When given a document, someone will ask a question, and someone else gives an answer. Then based on the answer, a related question is asked for deeper understanding. This process is performed iteratively, which can be regarded as a multi-turn conversation. After incorporating conversation into MRC, conversational machine reading comprehension (CMRC) has become the research hot spot.

In contrast to MRC, conversation history H also acts as part of the context to help with answer prediction in CMRC. The definition of CMRC can be formulated as shown in Table 14:

Table 14. Definition of conversational machine reading comprehension.

Conversational Machine Reading Comprehension
Given the context C, the conversation history with previous questions and answers $H = \{Q_1, A_1, \cdots, Q_{i-1}, A_{i-1}\}$ and the current question Q_i, the CMRC task is to predict the correct answer A_i by learning the function $\mathcal{F}$ such that $A_i = \mathcal{F}(C, H, Q_i)$.

Many researchers have tried to create new datasets with given passages and a series of conversations to satisfy the requirements of CMRC tasks. Reddy et al. [99] release CoQA, a CMRC dataset with 8000 conversations about passages in seven domains. In CoQA, a questioner asks questions based on a given passage and an answerer gives answers, simulating a conversation between two humans when reading the passage. There is no limit to the answer forms in CoQA, which requires more contextual reasoning. Similarly, Choi et al. [100] introduce QuAC for question answering in context. Compared to CoQA, in QuAC, the passage is given only to the answerer, and the questioner asks the questions based on the title of the passages. The answerer answers the questions with subsequences of the original passage and determines whether the questioner can ask a follow-up question. Ma et al. [101] extend cloze test tasks to the conversational setting. They use dialogues between characters selected from transcripts of the TV show *Friends* to generate related context and ask to fill in the blanks with character names according to utterances and context. In contrast to the above two datasets, it is aimed at multi-party dialogue and pays attention to the doers of some actions. To illustrate the CMRC task more specifically, some examples in CoQA datasets are presented in Table 15.

Table 15. An example of the CoQA dataset

CMRC	
Passage:	Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.
Question 1:	Who had a birthday?
Answer 1 :	Jessica
Question 2:	How old would she be?
Answer 2 :	80
Question 3:	Did she plan to have any visitors?
Answer 3 :	Yes
Question 4:	How many?
Answer 4 :	Three
Question 5:	Who?
Answer 5:	Annie, Melanie and Josh

CMRC brings about some new challenges compared to MRC:

- Conversational History

In MRC tasks, questions and answers are based on given passages and questions are independent from the previous question-answering process. In contrast to that, conversational history plays an important role in CMRC. A follow-up question may be closely related to prior questions and answers. More specifically, as shown in Table 15, question 4 and 5 are relevant to question 3. Moreover, answer 3 can be a verification for answer 5. To face this challenge, dialogue pairs as conversational history are also fed to the CMRC systems as inputs.

- Coreference Resolution

Coreference resolution is a traditional task in natural language processing, and it is even more challenging in CMRC. The coreference phenomenon not only may occur in the context, but may appear in the question-and-answer sentences as well. Coreference can be sorted into two kinds, explicit and implicit. With explicit coreference, there are explicit markers, such as some personal pronouns. For instance, to answer question 1 *Who had a birthday* in Table 15, the model has to figure out that *"her"* in *Today was her birthday* refers to *Jessica*. Similarly, the understanding of question 2 is based on the knowledge that *she* means *Jessica*. By comparison, implicit coreference without explicit markers is much harder to figure out. Short questions with certain intentions that implicitly refer to previous content are a kind of implicit coreference. For example, to figure out the complete expression of question 4 (*How many are the visitors?*), the model should extract the correlation between question 4 and 3.

Over the past two years, some researchers have made an effort to tackle the above new challenges in CMRC tasks. Reddy et al. [99] propose a hybrid model, DrQA+PGNet, which combines the sequence-to-sequence and machine reading comprehension models to extract and generate answers. To integrate information on conversational history, they treat previous question-answer pairs as a sequence and append them to the context. Yatskar et al. [102] use an improved MRC model, Bi-DAF++ with ELMo [30] to answer questions based on the given context and conversational history. Rather than encoding previous dialogue information on the context representation, they label answers to previous questions in the context. Instead of simply concatenating previous question-answer pairs as inputs, Huang et al. [103] introduce a flow mechanism to deeply understand conversational history, which encodes hidden context representations during the process of answering previous questions. Similar to Reddy et al. [99], Zhu et al. [104] append previous question-answer pairs to current questions, but in order to find out related conversational history, they employ additional self-attention on questions.

CMRC tasks that incorporate conversation into MRC are in line with the general process that humans understand one thing in the real world. Although researchers have been aware of the significance of conversational information and succeeded in representing conversational history, there is little work on coreference resolution. If the coreference cannot be figured out correctly, it will result in performance degradation. The common coreference phenomena in CMRC make this task far more challenging.

6. Open Issues

Based on the analysis of the literature cited in this article, we observe that there are some open issues that remain unsolved in neural MRC, some of which may have been also discussed in related research, such as machine inference and open-domain QA. The most important issue in neural MRC is that the machine does not truly understand the given text, as existing models mainly rely on semantic matching to answer question. Experiments performed by Kaushik and Lipton [105] show that some MRC models perform unexpectedly well when being provided with just passages or questions. Although researchers have already made great strides in neural MRC, there is still a giant gap between MRC and human comprehension in the following aspects:

- Limitation of Given Context

Similar to reading comprehension on language proficiency tests, machine reading comprehension asks the machine to answer questions based on the given context. Such context is a necessity in MRC tasks, but restricts the application. In the real world, machines are not expected to help students with their reading comprehension exams but make question-answering or dialogue systems smarter. Efforts made in multi-passage MRC research have somewhat broken the limitation of given context, but there is still a long way to go as how to find the most relevant resources for MRC systems effectively determines the performance of answer prediction. It calls for a deeper combination of information retrieval and machine reading comprehension in the future.

- Robustness of MRC Systems

As Jia and Liang [106] point out, most existing MRC models based on word overlap are weak to adversarial question-answer pairs. For SQuAD, they add distracting sentences to the given context, which have semantic overlap with the question and might cause confusion but do not contradict the correct answer. With such adversarial examples, the performance of MRC systems drops dramatically, which reflects that machines cannot really understand natural language. Although the introduction of answer verification components can alleviate the side-effect of plausible answers in a way, the robustness of MRC systems should be enhanced to face the challenges of adversarial circumstances.

- Incorporation of External Knowledge

As an essential component of human intelligence, accumulated common sense or background knowledge is usually used in human reading comprehension. To mimic this, a knowledge-based MRC is proposed to improve the performance of machine reading with external knowledge. However, how to effectively introduce external knowledge and make full use of it remains an ongoing challenge. On the one hand, the structure of knowledge stored in knowledge bases is so different from the text in context and questions that it is difficult to integrate them. On the other hand, the performance of knowledge-based MRC is closely related to the quality of the knowledge base. Constructing of a knowledge base is time-consuming, requiring considerable human efforts. In addition, knowledge in knowledge bases is sparse, and most of the time, relevant external knowledge cannot be found directly to support answer prediction and further reasoning is required. The effective fusion of knowledge graphs and machine reading comprehension needs to be further investigated.

- Lack of Inference Ability

As mentioned before, most existing MRC systems are mainly based on semantic matching between the context and the question to give the answer, which results in MRC models being incapable of reasoning. As an example given by Liu et al. [107] shows, given the context that *five people on board and two people on the ground died*, the machine cannot infer the correct answer *seven* to the question *how many people died* because of the lack of inference ability. How to enable the machine with inference ability still requires further research.

- Difficulty in Interpretation

Although existing MRC systems excel in a variety of tasks, they still work in a black box manner, i.e., no information is provided about how exactly they predict the answer. The lack of interpretability is a major drawback in applications such as healthcare, where the rationale for a model's output is a requirement for trust. However, the release of some MRC datasets such as HotpotQA [108] may contribute to the interpretability of MRC systems. HotpotQA, consisting of questions requiring inference over multiple supporting documents, labels sentences useful for reasoning so that the MRC systems are allowed to explain the predictions with such information. In addition, with development in explainable artificial intelligence (XAI) [109–111], the MRC research field has good potential for improving the trust and transparency of MRC systems and in turn making applications using MRC techniques more practical.

7. Conclusions

This article has presented a comprehensive survey on the progress of neural machine reading comprehension. Based on a thorough analysis of recent work, we give the specific definitions of MRC tasks and compare them in depth. The general architecture of neural MRC models is decomposed into four modules, and prominent approaches used in each module are introduced in detail. Representative datasets as well as evaluation metrics are described according to different MRC tasks. In addition, considering the limitations of MRC, we shed light on some new trends and discuss open issues in this research field.

Author Contributions: Conceptualization, S.L., S.Z. and X.Z.; Formal analysis, S.L., S.Z. and X.Z.; Investigation, S.L.; Methodology, S.L. and S.Z.; Supervision, H.W. and W.Z.; Writing—original draft, S.L.; Writing—review & editing, S.L., S.Z. and X.Z.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Glossary of Items Mentioned

Term	Definition
Curse of Dimensionality	A phenomenon that appears with high-dimensional data, where the computational complexity grows exponentially with the number of dimensions increasing.
Transfer Learning	A method of using knowledge from a related task that has already been learned to learn new tasks.
Coattention	A kind of bidirectional attention mechanism designed by Xiong et al. [17] which can attend to the context and question simultaneously.
Dynamic Decoder	A mechanism proposed by Xiong et al. [17] that is based on LSTMs, dynamically predicting the start and end positions of the answer.
Zero-Shot Setting	A setting where labeled examples for training are not enough.
Masked Language Model	A model designed by Devlin et al. [32], which randomly mask some tokens from the input, and requires predicting the masked word based on its context.
Next-Sentence Prediction	A task designed by Devlin et al. [32], judging whether sentence *B* follows sentence *A*.
Out-of-Vocabulary	Unknown words that appear in the training examples but not in the pre-defined vocabulary.
Fine-Grained Gating	A mechanism proposed by Yang et al. [44] which uses linguistic features, such as POS tags and name-entity tags, as a gate to control the amount of information flowing to word-level and character-level representations.
Gradient Explosion	A problem where a considerable gradient accumulate as it moves backward through the layers while training a deep neural network.
Gradient Vanishing	A problem where the gradient tends to get smaller as it moves backward through the layers while training a deep neural network.
Multi-Head Attention	A mechanism proposed by Vaswani et al. [41] using different linear projections to project queries, keys and values more than once when performing attention function.
Self-Attention	Also *Intra-Attention*, an attention mechanism computing attention weights in a single sequence to highlight importance of different positions in the sequence.
Residual Connections	A mechanism that skip some layer connections to facilitate single propagation and mitigate gradient degradation.
Back-Propagation	A shortened form of *backward propagation of errors*, a method of training neural networks with errors propagating backwards.
Textual Entailment	A task to judge the relationship between text fragments whether a truth of one text fragment follows from another.

Term	**Definition**
Pointer Networks	A neural architecture proposed by Vinyals et al. [56] which uses attention mechanism as a pointer to select several discrete tokens in input sequence as the output.
Maxout Networks	A model proposed by Goodfellow et al. [57] whose output is the maximum of a set of inputs, to improve optimization and model averaging with dropout.
Highway Networks	An architecture proposed by Srivastava et al. [58] which contains information highways to accelerate the training process of gradient -based deep neural networks.
Multi-Task Learning	A kind of transfer learning which solves multiple tasks at the same time.
Key-Value Memory Networks	A neural network proposed by Miller et al. [85] which stores memory as key-value pairs for easy access.
Data Enrichment	The process of enriching the amount of raw data.

References

1. Lehnert, W.G. The Process of Question Answering. Ph.D. Thesis, Yale University, New Haven, CT, USA, 1977.
2. Hirschman, L.; Light, M.; Breck, E.; Burger, J.D. Deep Read: A Reading Comprehension System. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, MD, USA, 20–26 June 1999; Association for Computational Linguistics: Stroudsburg, PA, USA, 1999; pp. 325–332.
3. Riloff, E.; Thelen, M. A Rule-Based Question Answering System for Reading Comprehension Tests. In Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-based Language Understanding Sytems—Volume 6, Seattle, WA, USA, 4 May 2000; Association for Computational Linguistics: Stroudsburg, PA, USA, 2000; pp. 13–19.
4. Poon, H.; Christensen, J.; Domingos, P.; Etzioni, O.; Hoffmann, R.; Kiddon, C.; Lin, T.; Ling, X.; Ritter, A.; Schoenmackers, S.; et al. Machine Reading at the University of Washington. In Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, Los Angeles, CA, USA, 6 June 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 87–95.
5. Hermann, K.M.; Kočiskỳ, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems—Volume 1, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 1693–1701.
6. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2383–2392.
7. Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; Deng, L. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. *arXiv* **2016**, arXiv:1611.09268.
8. Qiu, B.; Chen, X.; Xu, J.; Sun, Y. A Survey on Neural Machine Reading Comprehension. *arXiv* **2019**, arXiv:1906.03824.
9. Chen, D. Neural Reading Comprehension and Beyond. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2018.
10. Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E. RACE: Large-Scale ReAding Comprehension Dataset from Examinations. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 785–794.
11. Sukhbaatar, S.; Weston, J.; Fergus, R.; Szlam, A. End-to-End Memory Networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2015; pp. 2440–2448.
12. Chen, D.; Bolton, J.; Manning, C.D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 2358–2367.

13. Kadlec, R.; Schmid, M.; Bajgar, O.; Kleindienst, J. 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), Berlin, Germany, 7–12 August 2016; pp. 908–918.
14. Sordoni, A.; Bachman, P.; Trischler, A.; Bengio, Y. Iterative Alternating Neural Attention for Machine Reading. *arXiv* **2016**, arXiv:1606.02245.
15. Wang, S.; Jiang, J. Machine Comprehension Using Match-LSTM and Answer Pointer. *arXiv* **2016**, arXiv:1608.07905.
16. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional Attention Flow for Machine Comprehension. *arXiv* **2016**, arXiv:1611.01603.
17. Xiong, C.; Zhong, V.; Socher, R. Dynamic Coattention Networks for Question Answering. *arXiv* **2016**, arXiv:1611.01604.
18. Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; Hu, G. 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), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 593–602.
19. Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W.; Salakhutdinov, R. Gated-Attention Readers for Text Comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1832–1846.
20. Pan, B.; Li, H.; Zhao, Z.; Cao, B.; Cai, D.; He, X. MEMEN: Multi-Layer Embedding with Memory Networks for Machine Comprehension. *arXiv* **2017**, arXiv:1707.09098.
21. Wang, W.; Yang, N.; Wei, F.; Chang, B.; Zhou, M. 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), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 189–198.
22. Xiong, C.; Zhong, V.; Socher, R. DCN+: Mixed Objective and Deep Residual Coattention for Question Answering. *arXiv* **2017**, arXiv:1711.00106.
23. Tan, C.; Wei, F.; Yang, N.; Du, B.; Lv, W.; Zhou, M. S-Net: From Answer Extraction to Answer Synthesis for Machine Reading Comprehension. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
24. McCann, B.; Bradbury, J.; Xiong, C.; Socher, R. Learned in Translation: Contextualized Word Vectors. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 6294–6305.
25. Yu, A.W.; Dohan, D.; Luong, M.T.; Zhao, R.; Chen, K.; Norouzi, M.; Le, Q.V. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv* **2018**, arXiv:1804.09541.
26. Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; Zhou, M. Reinforced Mnemonic Reader for Machine Reading Comprehension. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; AAAI Press: Palo Alto, CA, USA, 2018, pp. 4099–4106.
27. Chen, Z.; Cui, Y.; Ma, W.; Wang, S.; Hu, G. Convolutional Spatial Attention Model for Reading Comprehension with Multiple-Choice Questions. *arXiv* **2018**, arXiv:1811.08610.
28. Chaturvedi, A.; Pandit, O.; Garain, U. CNN for Text-Based Multiple Choice Question Answering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 272–277.
29. Zhu, H.; Wei, F.; Qin, B.; Liu, T. Hierarchical Attention Flow for Multiple-Choice Reading Comprehension. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
30. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365.
31. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; Technical Report; OpenAI: San Francisco, CA, USA, 2018.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 3–5 June 2019; pp. 4171–4186.
33. Dhingra, B.; Liu, H.; Salakhutdinov, R.; Cohen, W.W. A Comparative Study of Word Embeddings for Reading Comprehension. *arXiv* **2017**, arXiv:1703.00993.
34. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA; Addison-Wesley: Harlow, UK, 2011.

35. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533. [CrossRef]
36. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013, pp. 3111–3119.
37. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
38. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
39. Nakov, P.; Ritter, A.; Rosenthal, S.; Sebastiani, F.; Stoyanov, V. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 1–18.
40. Clark, C.; Gardner, M. Simple and Effective Multi-Paragraph Reading Comprehension. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 845–855.
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017, pp. 5998–6008.
42. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multi-Task Learners. *OpenAI Blog* **2019**, *1*, 8.
43. Wang, Z.; Mi, H.; Hamza, W.; Florian, R. Multi-Perspective Context Matching for Machine Comprehension. *arXiv* **2016**, arXiv:1612.04211.
44. Yang, Z.; Dhingra, B.; Yuan, Y.; Hu, J.; Cohen, W.W.; Salakhutdinov, R. Words or Characters? Fine-Grained Gating for Reading Comprehension. *arXiv* **2016**, arXiv:1611.01724.
45. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1870–1879.
46. Zhang, J.; Zhu, X.; Chen, Q.; Dai, L.; Wei, S.; Jiang, H. Exploring Question Understanding and Adaptation in Neural-Network-Based Question Answering. *arXiv* **2017**, arXiv:1703.04617.
47. Hochreiter, S.; Schmidhuber, J. Long Sort-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
48. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
49. Luong, M.T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-Based Neural Machine Translation. *arXiv* **2015**, arXiv:1508.04025.
50. Rush, A.M.; Chopra, S.; Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. *arXiv* **2015**, arXiv:1509.00685.
51. Wang, Y.; Huang, M.; Zhao, L. Attention-Based LSTM for Aspect-Level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016, pp. 606–615.
52. Cheng, H.; Fang, H.; He, X.; Gao, J.; Deng, L. Bi-Directional Attention with Agreement for Dependency Parsing. *arXiv* **2016**, arXiv:1608.02076.
53. Cui, Y.; Liu, T.; Chen, Z.; Wang, S.; Hu, G. Consensus Attention-Based Neural Networks for Chinese Reading Comprehension. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 1777–1786.
54. Weston, J.; Chopra, S.; Bordes, A. Memory Networks. *arXiv* **2014**, arXiv:1410.3916.
55. Yu, S.; Indurthi, S.R.; Back, S.; Lee, H. A Multi-Stage Memory Augmented Neural Network for Machine Reading Comprehension. In Proceedings of the Workshop on Machine Reading for Question Answering, Melbourne, Australia, 19 July 2018; pp. 21–30.
56. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2692–2700.
57. Goodfellow, I.J.; Warde-Farley, D.; Mirza, M.; Courville, A.; Bengio, Y. Maxout Networks. *arXiv* **2013**, arXiv:1302.4389.
58. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks. *arXiv* **2015**, arXiv:1505.00387.

59. Shen, Y.; Huang, P.S.; Gao, J.; Chen, W. Reasonet: Learning to Stop Reading in Machine Comprehension. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; ACM: New York, NY, USA, 2017; pp. 1047–1055.
60. Trischler, A.; Ye, Z.; Yuan, X.; Bachman, P.; Sordoni, A.; Suleman, K. Natural Language Comprehension with the EpiReader. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 128–137.
61. Yu, Y.; Zhang, W.; Hasan, K.; Yu, M.; Xiang, B.; Zhou, B. End-to-End Answer Chunk Extraction and Ranking for Reading Comprehension. *arXiv* **2016**, arXiv:1610.09996.
62. Min, S.; Zhong, V.; Socher, R.; Xiong, C. Efficient and Robust Question Answering from Minimal Context over Documents. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1725–1735.
63. Suster, S.; Daelemans, W. CliCR: A Dataset of Clinical Case Reports for Machine Reading Comprehension. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 1551–1563.
64. Onishi, T.; Wang, H.; Bansal, M.; Gimpel, K.; McAllester, D. Who did What: A Large-Scale Person-Centered Cloze Dataset. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2230–2235.
65. Saha, A.; Aralikatte, R.; Khapra, M.M.; Sankaranarayanan, K. DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1683–1693.
66. Hill, F.; Bordes, A.; Chopra, S.; Weston, J. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv* **2015**, arXiv:1511.02301.
67. Bajgar, O.; Kadlec, R.; Kleindienst, J. Embracing Data Abundance: Booktest Dataset for Reading Comprehension. *arXiv* **2016**, arXiv:1610.00956.
68. Paperno, D.; Kruszewski, G.; Lazaridou, A.; Pham, N.Q.; Bernardi, R.; Pezzelle, S.; Baroni, M.; Boleda, G.; Fernandez, R. The LAMBADA Dataset: Word Prediction Requiring a Broad Discourse Context. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1525–1534.
69. Xie, Q.; Lai, G.; Dai, Z.; Hovy, E. Large-Scale Cloze Test Dataset Created by Teachers. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2344–2356.
70. Richardson, M.; Burges, C.J.; Renshaw, E. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 193–203.
71. Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; Suleman, K. NewsQA: A Machine Comprehension Dataset. In Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, BC, Canada, 3 August 2017; pp. 191–200.
72. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 784–789.
73. Joshi, M.; Choi, E.; Weld, D.; Zettlemoyer, L. 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), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1601–1611.
74. Weston, J.; Bordes, A.; Chopra, S.; Rush, A.M.; van Merriënboer, B.; Joulin, A.; Mikolov, T. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv* **2015**, arXiv:1502.05698.
75. Dunn, M.; Sagun, L.; Higgins, M.; Guney, V.U.; Cirik, V.; Cho, K. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv* **2017**, arXiv:1704.05179.
76. Kočiskỳ, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K.M.; Melis, G.; Grefenstette, E. The NarrativeQA Reading Comprehension Challenge. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 317–328. [CrossRef]

77. He, W.; Liu, K.; Liu, J.; Lyu, Y.; Zhao, S.; Xiao, X.; Liu, Y.; Wang, Y.; Wu, H.; She, Q.; et al. DuReader: A Chinese Machine Reading Comprehension Dataset from Real-world Applications. In Proceedings of the Workshop on Machine Reading for Question Answering, Melbourne, Australia, 19 July 2018; pp. 37–46.
78. Lin, C.Y. Rouge: A Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004.
79. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 311–318.
80. Ostermann, S.; Modi, A.; Roth, M.; Thater, S.; Pinkal, M. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan, 7–12 May 2018.
81. Long, T.; Bengio, E.; Lowe, R.; Cheung, J.C.K.; Precup, D. World Knowledge for Reading Comprehension: Rare Entity Prediction with Hierarchical LSTMs Using External Descriptions. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 825–834.
82. Yang, B.; Mitchell, T. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1436–1446.
83. Mihaylov, T.; Frank, A. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 821–832.
84. Sun, Y.; Guo, D.; Tang, D.; Duan, N.; Yan, Z.; Feng, X.; Qin, B. Knowledge Based Machine Reading Comprehension. *arXiv* **2018**, arXiv:1809.04267.
85. Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.H.; Bordes, A.; Weston, J. Key-Value Memory Networks for Directly Reading Documents. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1400–1409.
86. Wang, C.; Jiang, H. Explicit Utilization of General Knowledge in Machine Reading Comprehension. In Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2263–2272.
87. Levy, O.; Seo, M.; Choi, E.; Zettlemoyer, L. Zero-Shot Relation Extraction via Reading Comprehension. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017, pp. 333–342.
88. Tan, C.; Wei, F.; Zhou, Q.; Yang, N.; Lv, W.; Zhou, M. I Know There Is No Answer: Modeling Answer Validation for Machine Reading Comprehension. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Hohhot, China, 26–30 August 2018; Springer: Cham, Switzerland, 2018; pp. 85–97.
89. Hu, M.; Wei, F.; Peng, Y.; Huang, Z.; Yang, N.; Li, D. Read+ Verify: Machine Reading Comprehension with Unanswerable Questions. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6529–6537.
90. Sun, F.; Li, L.; Qiu, X.; Liu, Y. U-Net: Machine Reading Comprehension with Unanswerable Questions. *arXiv* **2018**, arXiv:1810.06638.
91. Dhingra, B.; Mazaitis, K.; Cohen, W.W. Quasar: Datasets for Question Answering by Search and Reading. *arXiv* **2017**, arXiv:1707.03904.
92. Htut, P.M.; Bowman, S.R.; Cho, K. Training a Ranking Function for Open-Domain Question Answering. In Proceedings of the NAACL HLT 2018, New Orleans, LA, USA, 2–4 June 2018; p. 120.
93. Lee, J.; Yun, S.; Kim, H.; Ko, M.; Kang, J. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 565–569.
94. Wang, S.; Yu, M.; Guo, X.; Wang, Z.; Klinger, T.; Zhang, W.; Chang, S.; Tesauro, G.; Zhou, B.; Jiang, J. R 3: Reinforced Ranker-Reader for Open-Domain Question Answering. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

95. Das, R.; Dhuliawala, S.; Zaheer, M.; McCallum, A. Multi-Step Retriever-Reader Interaction for Scalable Open-domain Question Answering. *arXiv* **2019**, arXiv:1905.05733.
96. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Su, L.; Cheng, X. HAS-QA: Hierarchical Answer Spans Model for Open-domain Question Answering. *arXiv* **2019**, arXiv:1901.03866.
97. Lin, Y.; Ji, H.; Liu, Z.; Sun, M. Denoising Distantly Supervised Open-Domain Question Answering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1736–1745.
98. Wang, S.; Yu, M.; Jiang, J.; Zhang, W.; Guo, X.; Chang, S.; Wang, Z.; Klinger, T.; Tesauro, G.; Cambell, M. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. *arXiv* **2017**, arXiv:1711.05116.
99. Reddy, S.; Chen, D.; Manning, C.D. CoQA: A Conversational Question Answering Challenge. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 249–266. [CrossRef]
100. Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.t.; Choi, Y.; Liang, P.; Zettlemoyer, L. QuAC: Question Answering in Context. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2174–2184.
101. Ma, K.; Jurczyk, T.; Choi, J.D. Challenging Reading Comprehension on Daily Conversation: Passage Completion on Multiparty Dialog. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2039–2048.
102. Yatskar, M. A Qualitative Comparison of CoQA, SQuAD 2.0 and QuAC. 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), Minneapolis, MN, USA, 3–5 June 2019; pp. 2318–2323.
103. Huang, H.Y.; Choi, E.; Yih, W.t. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. *arXiv* **2018**, arXiv:1810.06683.
104. Zhu, C.; Zeng, M.; Huang, X. SDNet: Contextualized Attention-Based Deep Network for Conversational Question Answering. *arXiv* **2018**, arXiv:1812.03593.
105. Kaushik, D.; Lipton, Z.C. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 5010–5015.
106. Jia, R.; Liang, P. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2021–2031.
107. Liu, S.; Zhang, S.; Zhanga, X.; Wang, H. R-Trans: RNN Transformer Network for Chinese Machine Reading Comprehension. *IEEE Access* **2019**, *7*, 27736–27745. [CrossRef]
108. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2369–2380.
109. Shrikumar, A.; Greenside, P.; Shcherbina, A.; Kundaje, A. Not Just a Black Box: Learning Important Features through Propagating Activation Differences. *arXiv* **2016**, arXiv:1605.01713.
110. Lipton, Z.C. The Mythos of Model Interpretability. *arXiv* **2016**, arXiv:1606.03490.
111. Montavon, G.; Lapuschkin, S.; Binder, A.; Samek, W.; Müller, K.R. Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition. *Pattern Recognit.* **2017**, *65*, 211–222. [CrossRef]