



Deep learning-based question answering: a survey

Heba Abdel-Nabi¹ · Arafat Awajan^{1,2} · Mostafa Z. Ali³

Received: 12 August 2021 / Revised: 17 October 2022 / Accepted: 23 October 2022 /

Published online: 30 December 2022

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Question Answering is a crucial natural language processing task. This field of research has attracted a sudden amount of interest lately due mainly to the integration of the deep learning models in the Question Answering Systems which consequently power up many advancements and improvements. This survey aims to explore and shed light upon the recent and most powerful deep learning-based Question Answering Systems and classify them based on the deep learning model used, stating the details of the used word representation, datasets, and evaluation metrics. It aims to highlight and discuss the currently used models and give insights that direct future research to enhance this increasingly growing field.

Keywords Question answering · Natural Language Processing · Reading comprehension · Deep learning · Convolutional network · Recurrent network · Attention mechanism

1 Introduction

The availability of huge amounts of data traveled through the networks and shared between users across the web and precisely through social media daily raised the need to develop automatic methods and approaches to process and represent this data. Most of these data have the form of natural language that is often semi-structured or even unstructured. This led to the development of a computational subset from the Artificial Intelligence (AI) field that aims to develop a machine that owns a human-like language processing ability in order to be able to read, represent, analyze, understand and reason over natural language text, which is proven to be a difficult task.

✉ Heba Abdel-Nabi
heb20179004@std.psut.edu.jo

Arafat Awajan
awajan@psut.edu.jo

Mostafa Z. Ali
mzali@just.edu.jo

¹ Department of Computer Science, Princess Sumaya University for Technology, Amman, Jordan

² Computer Science Department, Mutah University, Karak, Jordan

³ Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan

The aforementioned big data availability introduces complexity, data sparsity, and uncertainty that needs to be addressed. Therefore, one of the most challenging tasks of Natural Language Processing (NLP) is Question Answering, and it can be thought of as an umbrella under which all the NLP tasks can be accomplished since every NLP task can be formulated as a Question Answering task, such as sentiment analysis, text summarization, and document classification.

Due to the importance of the question answering task, it witnesses an increasing number of research works proposed to tackle its challenges. Some recent question answering surveys are available in the literature. However, these surveys are either general but cover a minimal number of state-of-the-art works that do not represent the entire field of question answering, such as the survey in [1]; or they are specific focusing only on certain question answering types, such as the survey in [2] that investigated Community Question Answering systems, the survey in [3] that was concerned with Machine Reading Comprehension Question Answering Systems, the survey in [4] that tackled the Open-domain Question Answering Systems and classified the revised papers according to the enhancement done to each of the component modules, or the survey in [5] that categorized the sequence-to-sequence generative models. The latest survey and most related to this work is the survey in [6]; however, our survey discusses more papers and provides an in-depth analysis of the used techniques in addition to classification based on their deep learning category and including information about their tested datasets, their used word encodings, and their adopted evaluation metrics, unlike what was done in [6] that only explain these categories briefly without any classification. Furthermore, this work refreshes [6] by adding and discussing the recently published papers to accommodate for the rapid development in the field to shape its current progress.

Our survey offers a comprehensive review and extensive study of the latest state-of-the-art question answering systems adopting deep learning techniques. This survey provides a new taxonomy of question answering systems from multiple criteria based on the used context, the answer, the domain, and the reasoning that characterize these systems. Also, the famous datasets based on this taxonomy are categorized. Moreover, our survey proposes another classification of question answering systems based on the deep learning category to provide more insights into the research directions in this field and distinguish the most probable category to adopt for future research. In particular, we divide the reviewed systems into ten categories and provide a summarization of each work in each category. Furthermore, a brief explanation and comparison of the recent language models is made to highlight the potential choices to adopt the enhancement of the contextualized representations needed to encode the text in future systems. Also, we discuss the specifications of the publicly available popular Question Answering datasets and the characteristics of the evaluation metrics essential in assessing the performance and validating the results of any question answering system. Moreover, we collected the results of some of the reviewed models that used those datasets for evaluation to give a fair base for comparison. Thus, we provide our analysis according to different points of view and provide some statistics to pay attention to the current research directions in this area. Our aim is to guide interested beginners and expert researchers to better explore the field.

We particularly seek to answer the following research questions to provide a vital understanding of the current stand of Deep learning-based Question Answering research in order to propose new avenues:

- What is the current state-of-the-art Question Answering systems that utilize deep learning as their base?
- What are the main challenges and gaps that face the Question Answering field?

- What are the future directions to enhance the existing Question Answering systems?

Since Deep Learning is an active area of research that changes rapidly, we limit our examination and analysis of the research papers in the time interval after 2015 to the latest, as far as we know. Our aim is to establish a strong foundation and a general understanding to aid the researchers who are interested in this area. Our survey provides a panoramic view of the entire textual deep learning-based question answering research field that begins with the early immature works on simple extractive deep learning-based question answering systems and exploits up to the recent generative and multi-hop reasoning-based systems.

This survey is organized as follows: in the second section, a short discussion of Question Answering systems, in general, is given, followed by the main categories of the Question Answering systems and a brief introduction to deep learning. After that, a high-level description of the word embedding techniques and the evaluation metric used in the field are presented. A classification and summarization of the recent research work according to the used deep model with specifying the details of the used word representation, the datasets, and the evaluation metrics are presented in section three. In section four, an explanation of the most popular Question Answering datasets along with some statistics and performance results is given. The discussion of the major directions and methods used is given in section five. Finally, in section six, the conclusion and future directions are outlined. Figure 1 illustrates the remaining structure of the survey.

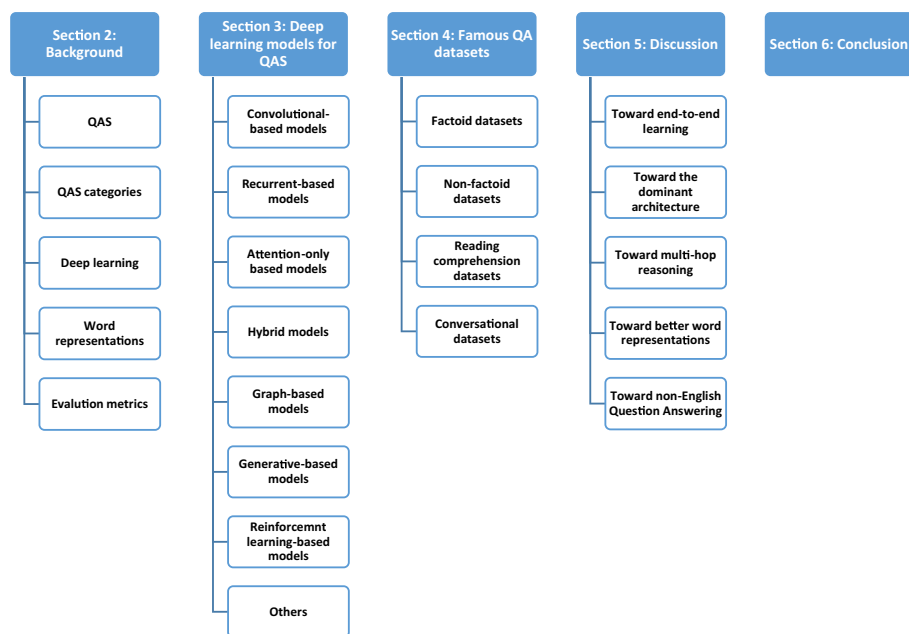


Fig. 1 The structure of the paper

2 Background

2.1 Question answering system (QAS)

Question Answering (QA) is a complex and challenging task in NLP that requires natural language understanding, natural language generation, and world knowledge representation. This is done to automatically provide the appropriate answer to a certain question asked in natural language by deducing the most relevant answers from a Knowledge Base (KB). QA can be viewed as a tool to assist the degree of understanding the machine obtained toward a certain text [7] by its ability to answer related questions. It is a sort of advanced information retrieval. The two main difficulties of the QAS are bridging the semantic gap between the question and the answer pair and reasoning over the relationships in the facts [8].

The classical QAS, as depicted in Fig. 2, consists of three components. The first is the question processing and classification module that analyzes and defines the question focus using question-type classification or pattern-matching rules. It may sometimes do a query expansion where semantically equivalent multiple questions are generated. The information retrieval or document processing module is responsible for retrieving a ranked set of relevant documents based on the given question. This module aims to shorten the search space from whole documents into paragraphs; thus, this module is sometimes called “paragraph indexing.” Finally, the Answer Selection module extracts and validates the correct answer using similarity measures.

The designing of a Question Answering system has undergone two main traditional approaches [9]; the first approach is the rule-based system, where these rules are constructed from the generated lexical and semantic features in the questions [10]; however, this approach is fragile since these rules are not followed strictly for the spoken and written natural language texts. Famous examples of a QAS based on this approach are [11–13]. The second approach is the statistical and probabilistic data-driven approaches, such as the work in [14]; nevertheless, this approach fails to deal with synonyms or words outside the training dataset, i.e., Out-Of-Vocabulary (OOV) words. This raises the potential advantages of using deep learning-based models in the QAS. Figure 3 gives an overview of the general architecture of deep learning-based QAS. It consists of three main components. The first is the context processing and retrieval to extract the most relevant context to the question, either from the provided corpus or from an external KB or commonsense source. This component is optional and only applicable in retrieval-based QASs, as discussed in the next section. The second

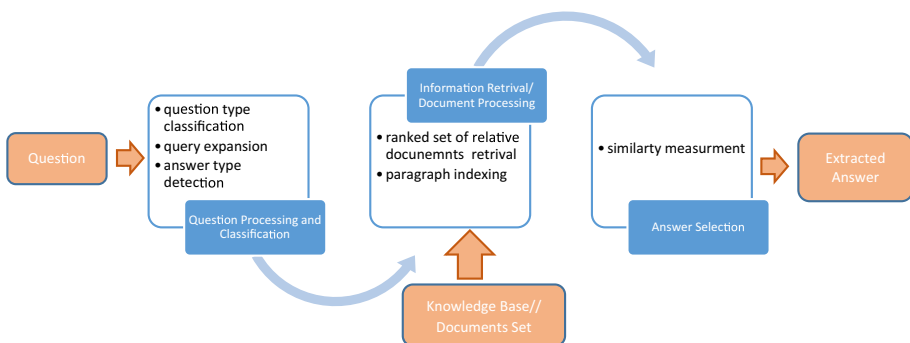


Fig. 2 The classical QAS pipeline

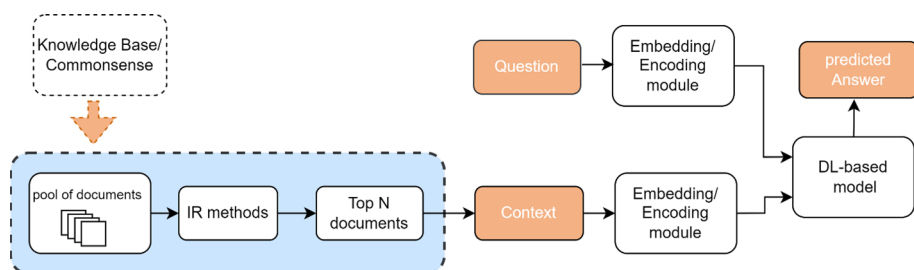


Fig. 3 The deep learning QAS pipeline. The dashed line means an optional component

component requires generating vector representations of the context and the question to be fed to the third component, which is a deep learning-based model. Finally, the answer is predicted and decoded, and then evaluated. This type of QAS is usually end-to-end without requiring any feature engineering.

2.2 QAS categories

The research in the QAS field is huge and diverse. By analyzing the proposed works, QAS can be classified into different categories depending on multiple factors, such as context type, context source, reasoning type, answer format and type, and domain. Figure 4 shows

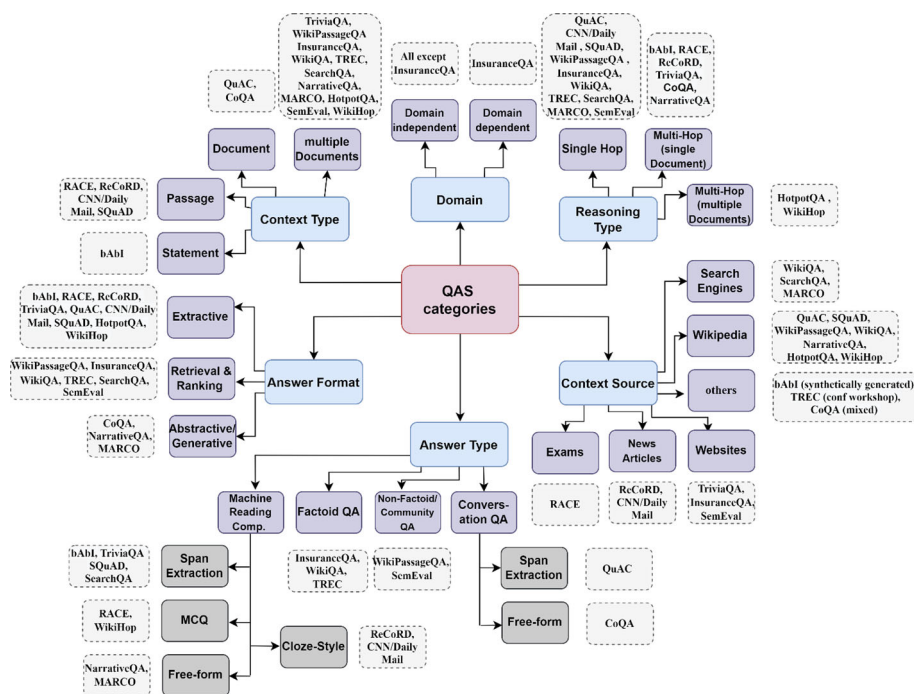


Fig. 4 Taxonomy of QAS categories with names of example datasets representative of each category

the taxonomy we adopted to categorize the QAS based on these factors and examples from the datasets reviewed in this survey that belongs to each category.

1. **Context type;** the context is the input provided to extract the answer to the asked question. It can be consecutive statements, a passage (single paragraph), a single document with multiple paragraphs, or multiple documents. The answer prediction difficulty increases gradually from statements to multiple documents because of the associated noise and the growing search space that may require multiple hops to reach the answer.
2. **Context source;** most of the context from which the answer should be predicted comes from Wikipedia articles, usually equipped with search engines' logs. Other websites like news are also used. Some contexts are extracted from English exams, and the other is synthetically generated to suit the task at hand.
3. **Reasoning type;** to find the correct answer to a given question and assist the ability of QAS to understand and analyze the text of the given context, single or multiple passes over a single or multiple paragraphs in one or multi-documents are performed. Therefore, we classify the reasoning type of QAS into a single hop, multiple hops within a single document, and multiple hops over multiple documents. The latter is the most complex and challenging type since it requires gathering the scattered supporting facts from several documents and jointly comprehending their relationships with the question to interpret the appropriate answer.
4. **Answer format;** the nature of the answer in QAS can be classified into three major types; extractive, retrieval and ranking, and abstractive. In QAS with an extractive answer, the answer words are explicitly found and selected on the context, and usually, the answer is a span. The second type requires extracting multiple statements from the retrieved context passages and then ranking them to select the most relevant answer spans to the question; this type is usually adopted in the open domain QAS or community-based ones. At last, the third abstractive type generates a descriptive answer in natural language with correct semantic and lexical structure using words that are not necessarily limited to the context.
5. **Domain;** according to the available knowledge, QASs can be divided into closed and open domain QAS. The answer in the open domain systems (aka, domain-independent) is not limited to a specific domain. It depends on the general world knowledge and any provided knowledge on the dataset. Therefore, this reflects the realistic nature of this type of system. The vocabulary of the open domain systems is general and does not require any domain-specific terms. As a result, this requires the involvement of a huge amount of knowledge that must be processed. On the other hand, the closed domain systems (aka, domain-dependent) restrict the KB and the domain of questions; therefore, the resulting answers are exact and accurate and only obtained from the dataset's KB. It contains a specific domain terminology and vocabulary. Therefore, the answer quality in open-domain systems is low compared to the quality obtained from the closed-domain ones.
6. **Answer type;** the answer nature is subdivided into factoid, non-factoid, reading comprehension, and conversational question answering types.
7. **Factoid QAS** is based on retrieving short fact answers like single entities such as names, numbers or nouns or retrieving isolated sentences from this knowledge. An example of questions and corresponding answers of factoid datasets is given in Fig. 5.
8. **Non-factoid QAS** aims for answer passage retrieval. It is based on retrieving descriptions or definitions regarding the question. A huge section of this type of systems is the community-based systems where the questions are asked by community members (for

<p><i>Query 201:</i> Question: What was the name of the first Russian astronaut to do a spacewalk? Answer: Aleksei A. Leonov Answer Document ID: LA072490-0034</p> <p><i>Query 202:</i> Question: Where is Belize located? Answer: Central America Answer Document ID: FT934-14974</p>
--

Fig. 5 An example from the TREC dataset (obtained from [15])

example, search engines like yahoo, google,...). The single question gets more than one answer from the other users. The answer selection in this type of systems is to determine the answer's relevance to the user question. An example question and corresponding answer of a non-factoid dataset is given in Fig. 6.

9. Reading Comprehension System aims to find the question's correct answer by reasoning and analyzing a specific context. The answers from this type of systems are a mixture of factoid and non-factoid types. Examples of questions and corresponding answers from single and multi-hop reading comprehension datasets are given in Fig. 7, respectively. Reading comprehension QAS can be further subdivided into span extraction, in which the answer is a span of words in the context; Multiple Choice Questions (MCQ), in which the answer is one of the multiple candidate statements; cloze-style, in which the answer is a missing token from a sentence; and free-form (abstractive), in which the answer is generated in natural language.
10. Conversational QAS, the most challenging type, requires a contextual understanding of the dialogue history, i.e., the previous questions and their answers, in addition to the current passage and question. In other words, the questions in conversational QAS have a cyclic multi-turn nature; they are not stand-alone questions. Figure 8 gives example questions and corresponding answers in the conversational dataset. The QAS that belongs to this category can be extractive or abstractive.

2.3 Deep learning

Since the deep learning breakthrough in 2006 [18], several deep learning architectures and models have been proposed in the NLP field. Deep learning gives the power to eliminate the empirical choices of the representations and lexical, syntactic, and semantic features [19] selection that is used to form an accurate and efficient QAS because it is trained in an end-to-end fashion. Deep learning tries to mimic human ways of thinking and reasoning. It learns a hierarchical representation of the textual natural language without the need for manual feature engineering or the existence of resources such as lexical knowledge like WordNet [20] or linguistic knowledge such as grammar rules. This gives deep learning the capability of generalizing easily to new domains and languages. Deep learning can produce answers without relying on complex hand-crafted features that are considered time-consuming and domain-dependent. Even these hand-crafted features may fail to offer an overall representation of the input. These properties differentiate the deep learning methods from the traditional text manipulation and the classical machine learning methods and give them their success. Some applications of the usage of deep learning in the NLP field are: Named Entity Recognition (NER) [21], Text classifications [22], Text summarizations [23], and Machine Translation [24].

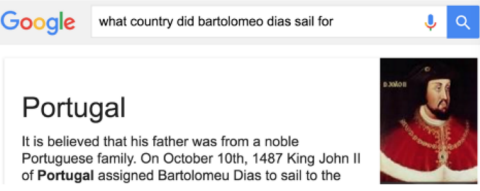
	
WIKISUGGEST Query	Answer
what year did virgina became a state	1788
general manager of smackdown	Theodore Long
minnesota viking colors	purple
coco martin latest movies	maybe this time
longest railway station in asia	Gorakhpur
son from modern family	Claire Dunphy
north dakota main religion	Christian
lands end' brand	Lands' End
wdsu radio station	WCBE

Fig. 6 An example of the WIKISUGGEST dataset [16]

Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula.

The atomic number of the periodic table for oxygen?
 • Ground Truth Answers: 8

What is the second most abundant element?
 • Ground Truth Answers: helium

How many atoms combine to form dioxygen?
 • Ground Truth Answers: two

Paragraph A, Return to Olympus:
 [1] *Return to Olympus* is the only album by the alternative rock band Malfunkshun. [2] It was released after the band had broken up and after lead singer Andrew Wood (later of Mother Love Bone) had died of a drug overdose in 1990. [3] Stone Gossard, of Pearl Jam, had compiled the songs and released the album on his label, Loosegroove Records.

Paragraph B, Mother Love Bone:
 [4] *Mother Love Bone* was an American rock band that formed in Seattle, Washington in 1987. [5] The band was active from 1987 to 1990. [6] Frontman Andrew Wood's personality and compositions helped to catapult the group to the top of the burgeoning late 1980s/early 1990s Seattle music scene. [7] Wood died only days before the scheduled release of the band's debut album, "Apple", thus ending the group's hopes of success. [8] The album was finally released a few months later.

Q: What was the former band of the member of Mother Love Bone who died just before the release of "Apple"?
A: Malfunkshun
Supporting facts: 1, 2, 4, 6, 7

Fig. 7 Examples of reading comprehension, the right (single-hop example from SQuAD v1.1 dataset) and the left (multi-hop reasoning example from HotPotQA dataset)

2.4 Word representation

In order to deal with the deep learning techniques, the textual representations of the input of the QAS must be converted into numerical representations that suit the deep neural models. Textual Embedding is a technique for language modeling and feature learning using vectors that encode linguistic regularities and patterns. It captures and models the semantic and syntactic relationships between the words and their context.

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. Jessica had . . .

Q₁: Who had a birthday?

A₁: Jessica

R₁: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q₂: How old would she be?

A₂: 80

R₂: she was turning 80

Q₃: Did she plan to have any visitors?

A₃: Yes

R₃: Her granddaughter Annie was coming over

Q₄: How many?

A₄: Three

R₄: 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.

Q₅: Who?

A₅: Annie, Melanie and Josh

R₅: 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.

Fig. 8 An example of the CoQA dataset, where Q is a question, A is an answer, and R is a rationale that supports the answer [17]

Different embedding approaches are used to obtain the word representations that can be grouped into two main categories:

2.4.1 Static/context-independent embeddings

Also known as word embeddings, such as Word2vec [25] and GLoVe [26]. These methods output a single embedding vector of each word regardless of the number of word occurrences in the text with different positions, meanings, or aspects, i.e., they suppress all the word meanings in one vector. The resulting word embedding vectors can be utilized directly since they are already trained on a large corpus.

- **Word2Vec**

Word2Vec¹ is proposed in [25]. It is a simple learning-based predictive architecture that aims to learn the features from the words in the corpus automatically. The Word2Vec not only reduces the high dimensionality of the word vector but also, at the same time, interacts with

¹ <https://github.com/tmikolov/word2vec>, Date of Access: 5th Jun, 2022.

the surroundings of the word, i.e., the word's context. This is achieved using two algorithms: Skip Gram (SG) or Continuous Words of Bag (CBOW) models. In the SG model, the input word to the neural network is used to predict its neighbor contexts, unlike the CBOW model, in which the input word itself is predicted using its surrounding contexts.

• GLoVe

GLoVe,² an abbreviation of **GL**ocal **VE**ctor, is proposed by [26] based on matrix factorization. It is a statistical unsupervised machine learning-based model that forms statistics of the count of overall vocabulary in the corpus and learns by reconstructing the co-occurrence matrix that initially reports the context words' counts. Then, this co-occurrence is compressed using a dimensionality reduction method that preserves its original probabilities.

2.4.2 Dynamic/context-dependent embeddings

Also known as Language Models (LMs), such as BERT [27], ELMo [28], LUKE [29], ELECTRA [30], XLNET [31], BIGBIRD [32], and ANNA [33], where the embedding vector for each word occurrence in the text captures its meaning at that particular position. Since these contextual embeddings are sensitive to the context (surrounding text), they must be re-trained and fine-tuned upon usage to reflect the current context. Table 1 compares these LMs in terms of the number of trainable parameters, the main method, the training data source, and the pre-train infrastructure used.

• BERT Family

BERT, an abbreviation of **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, is a recent model proposed in [27] and is built to generate dynamic word embedding based on the transformer encoder [34], which is a multi-layered bidirectional attention-based model with positional encodings. Thus, each word in BERT is represented using a single vector that reflects its content and position embeddings. BERT is trained to accomplish the masked language modeling objective; where a masked word is predicted using its surroundings, in addition to the next sentence prediction objective. BERT is divided based on its size into BERT-Base and BERT-Large. Moreover, BERT handles the OOV issue since it represents inputs as subwords instead of whole words, as done in Word2Vec and GLoVe, which balance the character-based and word-based representations. The huge trained dataset size, the deep architecture, and the use of subwords as input are all reasons to succeed in achieving a superior performance in capturing complex linguistic phenomena for better language understanding [35]. Many variants of the BERT are recently proposed to ease computational needs, such as ALBERT [36] and DistilBERT [37], or to boost performance, such as StructBERT [38], RoBERTa [39], DeBERTa [40] and ConvBERT [41].

A LITE BERT (ALBERT) [36] claims that increasing the model size continuously often causes challenges in training, represented by longer training times and memory limitations. Two-parameter reduction techniques are used in ALBERT to tackle these problems: factorized embedding parameterization and cross-layer parameter sharing techniques. The former decomposes the vocabulary-embedding matrix into two small matrices, while the latter limits the growth of the number of parameters as the depth of the network increases in size.

DistilBERT [37] is an approximated version of BERT trained in a student–teacher training fashion. Through knowledge distillation, the DistilBERT student network tries to emulate the BERT teacher network's behavior and knowledge using triple loss that accounts for

² <https://github.com/stanfordnlp/GloVe>, Date of Access: 5th Jun, 2022.

Table 1 Comparison between the different LMs

LM	Size (#parameters)	Main method	Training data source	Pre-train infrastructure	Main contribution
BERT (2019)	BERT _{BASE} : 110M BERT _{LARGE} : 340M	Multi-layer bidirectional Transformer performs unsupervised Masked LM and Next Sentence Prediction objectives	BOOKCORPUS and English Wikipedia with a total training data of 16 GB and 3.3 billion word corpus	BERT _{BASE} : 4 Cloud TPUs (16 TPU chips) BERT _{LARGE} : 16 Cloud TPUs (64 TPU chips) Training time: 4 days	A significant milestone in NLP
ALBERT (2019)	ALBERT _{BASE} : 12M ALBERT _{LARGE} : 18M ALBERT _{XLARGE} : 60M ALBERT _{XXLARGE} : 235M	BERT with two parameter-reduction techniques	Same as BERT training data (16 GB)	Cloud TPU V3. The TPUs used for training ranged from 64 to 512, depending on model size	ALBERT is 1.7 times faster than BERT due to the 18 times reduction in size with slightly worse performance
DistilBERT (2019)	66M	Distilled BERT with triple loss	Same as BERT training data (16 GB)	Eight 16 GB V100 GPUs Training time: 90 h	40% Smaller than BERT, 60% faster, that retains 97% of the language understanding capabilities
StructBERT (2019)	StructBERT _{BASE} : 110M StructBERT _{LARGE} : 340M	BERT with joint word and sentence structural objectives	Documents from English Wikipedia (2.5 billion words) and BookCorpus	64 Tesla V100 GPUs StructBERT _{BASE} training time: 38 h StructBERT _{LARGE} training time: 7 days	StructBERT is competitive with BERT on a variety of language tasks

Table 1 (continued)

LM	Size (#parameters)	Main method	Training data source	Pre-train infrastructure	Main contribution
RoBERTa (2019)	110M	BERT with multiple design choices and training strategies	BERT data (16 GB) + 63M English articles from Common Crawl News (76 GB) + Web Text from URLs on Reddit (38 GB)+STORIES from Common Crawl dataset (31 GB). Total training data ~160 GB	1024 32 GB Nvidia V100 GPUs Training time: 1 day	RoBERTa outperforms BERT and removed the Next Sentence Prediction task
DeBERTa (2021)	DeBERTa _{BASE} : 134 M	Decoding-enhanced BERT with disentangled attention	Wikipedia (12 GB)+ BookCorpus (6 GB) + OPENWEB-TEXT (38 GB) + STORIES (31 GB). Total filtered data size: 78 GB	96 V100 GPUs Training time: 20 days	Comparable with RoBERTa _{LARGE} and trained on half of its training data
ConvBERT (2020)	Base: 96M	BERT with Span-based Dynamic Convolution	OpenWebText (32 GB)	Not well optimized for GPU and TUP use yet	ConvBERT is lighter, has lower training costs, fewer model parameters and outperform BERT
LUKE (2020)	483M	Transformer-based with a novel entity-aware self-attention mechanism	Dec 2018 version of Wikipedia, with 3.5 billion words and 11M entity annotations	16 NVIDIA Tesla V100 GPUs Training time: 30 days	LUKE outperforms RoBERTa because of its larger number of pre-training steps
ELECTRA (2020)	Small:14M Base:110M	Transformer-based with replaced token detection objective	Same as BERT training data (16 GB)	ELECTRA _{SMALL} : 1 GPU for 4 days ELECTRA _{BASE} : 16 TPU v3 for 4 days	The performance of ELECTRA is equivalent to XLNet and RoBERTa but at one-fourth of their computing requirements

Table 1 (continued)

LM	Size (#parameters)	Main method	Training data source	Pre-train infrastructure	Main contribution
XLNet (2019)	Base: 110M Large: 340M	Autoregressive transformer with factorization order permutations	BooksCorpus/English Wikipedia (13 GB) + Giga5 (16 GB) + Clue Web (19 GB) + Common Crawl(110 GB). Total 32.89 Billion words	512 TPU v3 chips Training time: 5.5 days	XLNet outperforms BERT and RoBERTa since it address their limitation: neglecting the masked positions dependency and the fine-tune discrepancy
BigBird (2020)		Transformer with a sparse attention mechanism	Books, CC-News, Stories, Wikipedia	64 Google Cloud TPUs	Suitable for modeling long text 8 times longer than what BERT handle
ANNA (2022)	ANNA _{BASE} : 60M ANNA _{LARGE} : 550M	Neighbor-aware attention mechanism	Wikipedia(16 GB) + Colossal-Cleaned Common Crawl (730 GB) + Books3 (100 GB)+OpenWebText (62 GB). Total 145.6 billion tokens	ANNA _{LARGE} : 256 TPU v3 for 10 days ANNA _{BASE} : 64 TPU for 5 days	Competitive performance against BERT, ALBERT, RoBERTa, and XLNet models in the task QA

the language modeling, distillation, and cosine-distance. The DistilBERT architecture is a light version of BERT with the number of Transformer layers reduced to half and highly optimized operations. Thus, DistilBERT could be used for the mobile on-the-edge application for question answering.

To account for the missing handling of the underlying linguistic structures in BERT presentations and consequently to obtain different language understanding levels, **StructBERT** [38] incorporates two novel linearization auxiliary objectives into the pre-training. The word-level ordering structural objective is directed to leverage the sequential ordering of inner-sentence words. It is trained jointly with the masked LM objective by shuffling the masked tokens and predicting their correct order. On the other hand, the sentence-level structural objective focuses on modeling dependencies of the inter-sentence structures by swapping the sentences and predicting the previous and the next one. Thus, StructBERT explicitly models the correct structure of words in its contextualized representation in a bidirectional manner.

The **Robustly optimized BERT PreTraining approach (RoBERTa)** [39] addresses the problem of the optimal choice of the hyper-parameters or the design choices of the classical BERT model that affect the model performance. RoBERTa suggests modifications to BERT parameters: increases the number of epochs and the batch size; removes the next sentence prediction objective; trains on longer sequences; dynamically changes the training data masking pattern; and utilizes ten times the BERT data for training.

The **Decoding-enhanced BERT with disentangled attention (DeBERTa)** [40] introduces two novel techniques to improve BERT: disentangled attention mechanism and enhanced mask decoder. The attention weights are computed based on two vectors for each word in DeBERTa that encode its content and relative position using disentangled matrices. On the other hand, the words' absolute positions are incorporated in the enhanced mask decoder, along with the aggregated contextual embeddings of words, to predict the masked tokens for model pre-training.

ConvBERT [41] addresses the intrinsic redundancy issue resulting from computing the attention maps generated across the whole input sequence in BERT's multi-head self-attention by proposing a novel mixed attention block. ConvBERT uses self-attention and a novel span-based dynamic convolution to capture important global and local contextual dependencies efficiently. ConvBERT incorporates the design of bottleneck attention structure with depth-wise separable convolution that has adaptive kernels that act on the input-span level instead of the single-token level to utilize the different contexts dynamically.

• ELMo

ELMo, an abbreviation of **E**mbdings from **L**anguage **M**odels, is a deep contextualized word representation proposed in [28] that models the complex syntax and semantics characteristics of words along with their linguistic contexts. Each learned word vector in ELMo is a function of a linear combination of all the internal states of a deep bidirectional LM (biLM) pre-trained on a large corpus. Thus, these resultant context-dependent features produce rich word representations. The key difference between the BERT variants and ELMo is that the former are fine-tune-based models, while the latter is feature-based.

• LUKE

LUKE, an abbreviation of **L**anguage **U**nderstanding with **K**nowledge-based **E**mbdings, is proposed in [29] to provide pre-trained contextualized representations designed to address entity-related reasoning tasks. LUKE is based on a bidirectional transformer model with a novel entity-aware self-attention mechanism. It manipulates two types of input tokens: words and entities, independently. The advantage of obtaining the embeddings for the entities is to

make LUKE able to inspect and model the relationships between them. Thus, it selects the appropriate attention mechanism based on the token type. LUKE uses RoBERTa as a base pre-trained model to perform the masked language modeling on an entity-annotated corpus obtained from Wikipedia.

• ELECTRA

ELECTRA, an abbreviation of **E**fficiently **L**earning an **E**ncoder that **C**lassifies **T**oken **R**eplacements **A**ccurately, is proposed in [30]. It tweaks the conventional generator nature of the text encoder in the transformer and makes it a discrimination-based operation by proposing the “replaced token detection” self-supervised pre-training task. Therefore, ELECTRA trains two transformer models: generator and discriminator. The tokens in the input are replaced, by the generator, with other synthetic alternatives sampled from a certain distribution and then trained as a masked LM. Contrarily, the discriminator attempts to distinguish whether each sequence token is a replacement or an original token rather than directly predicting the identity of the replaced masks as done in BERT. This results in better performance because ELECTRA’s replaced token detection task processes all the input tokens rather than only the masked ones. Thus, ELECTRA eases the requirements of the needed large training data. Although it uses the two complementary models, ELECTRA does not have an adversarial nature that characterizes generative models.

• XLNet

The generalized autoregressive LM, XLNet, is proposed in [31] to address the neglected dependency in BERT’s masked positions. It leverages the advantages of the segment recurrence mechanism and relative encoding scheme of the autoregressive Transformer-XL model in the pre-training of the BERT autoencoder. The permutation language modeling objective in XLNet enables bidirectional contextual learning by training an autoregressive model on all possible permutations of the word in a sentence, then maximizing the expected log-likelihood of a sequence over all permutations of the factorization order. Furthermore, since XLNet does not rely on data corruption, thus BERT’s pre-train-fine-tune discrepancy issue, which exists because of the missing masking at the fine-tuning phase, is also handled by XLNet.

• BIGBIRD

To address the sequence length quadratic dependency issue of BERT, BIGBIRD is proposed in [32]. BIGBIRD is a transformer-based LM with a sparse linear dependency attention mechanism. Motivated by the graph sparsification methods, BIGBIRD computes attention in three directions: diagonal direction to capture local attention, sides direction to capture global attention, and in sparse random places for better approximation of the full-attention matrix.

• ANNA

ANNA, an abbreviation of **A**pproach of **N**oun-phrase-based language representation with **N**ighbor-aware **A**ttention, is a pre-trained LM proposed in [33] to enrich the understanding of extractive QAS. ANNA is a transformer-based model with a new neighbor-aware self-attention mechanism. ANNA predicts the contextualized representations of masked “whole-span” of noun phrases and words to capture the syntactic, lexical, and contextual information. It focuses on the neighboring tokens’ relationships by ignoring the influence of identical tokens by masking the diagonality in the attention matrix.

2.5 Evaluation metrics

The evaluation of the QAS performance is a critical step that determines if the QAS is trustworthy and whether it can predict correct and accurate answers for a question selected from the testing dataset. The following are some famous evaluation metrics used in the deep learning-based QAS.

- **F1 score/ F1-measure**: it is considered the harmonic mean of the precision and recall rate. In other words, the F1 score measures the overlapping between the words in the predicted answer against the corresponding words in the ground truth answer (the correct answer).
- **Mean Average Precision (MAP)**: it is concerned with all the correct answers' ranks by taking the average of their precisions.
- **Mean Reciprocal Rank (MRR)**: it measures the ability of the QAS to return a ranked list of answers, i.e., it is computed by taking the inverse of the rank of any correct answer.
- **Exact Match (EM)**: it measures the matching percentage between the potential predicted answer and the ground truth answer. The ideal case is when EM is 1, which indicates that the prediction is identical to the ground truth.
- **Accuracy**: it measures the percentage of correctly answered questions by the QAS, which consequently indicates the effectiveness and the level of trust the system offers.
- **Word Error Rate (WER)**: it measures the edit distance required to transform a predicted answer to the associated ground truth answer. In particular, WER determines the required number of insertions, deletions, substitutions, and transpositions. However, a major limitation of WER is that it heavily penalizes the different word ordering in the sentence.
- **Bilingual Evaluation Understudy (BLEU)**: it compares the lexical relationship between the generated answer and the ground truth answer by computing the n -gram overlap between them [42]; thus, it is a precision-based metric. BLEU aggregates the precisions of the generated answers of all the questions; therefore, it produces a corpus-level score. Usually, n values vary from 1 (BLEU-1) to 4 (BLEU-4) and are averaged geometrically to produce the final BLEU score.
- **Metric for Evaluation of Translation with Explicit Ordering (METEOR)**: it is proposed to overcome the BLEU's limitation of not considering the recall and performing exact n -gram matching [43]. METEOR, on the other hand, contains a relaxed matching strategy that consists of four stages: unigram mapping using the exact word, stemmed-word matching, synonym and paraphrase matching, and weighted F1-score computation. METEOR has a fragmentation penalty factor that rewards the longer contiguous alignments, i.e., longer n -gram matches. METEOR results in a good correlation with human judgments on the sentence level, as opposed to BLEU, which seeks correlation at the corpus level.
- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE)**: it has many variants to suit different NLP tasks, with ROUGE-N and ROUGE-L being the most popular variant to evaluate textual QAS [44]. ROUGE-N counts the n -gram matches between the free-form generated answer by the QAS and the ground truth answer by measuring the ratio of the number of overlapped n -grams to the total n -grams of the ground truth answer. It differs from BLEU-N by being a recall-based measure, unlike the latter, which is precision-based. On the other hand, ROUGE-L measures the longest matching common subsequence between the generated answer and the ground truth answer based on the F1 score. ROUGE-L checks for the in-sequence order of the matches, not for their consecutiveness.

Several works reported the inability of n -gram-based models, such as BLEU and ROUGE, to truly evaluate the generative abstractive answers since they rely on the exact lexical features

of the sentences and are sensitive to any variants in the word forms. Although METEOR, using its relaxed matching strategy, partially addresses some of these deficiencies, it faces another issue, i.e., the answers can have many semantic or syntactic variations; they can be phrased in different ways and with different word-wording structures. Therefore, many new automatic metrics have been proposed in recent years to evaluate the fluency and correctness of the generated answer, such as BERTscore [45] and KPQA [46].

- **BERTscore:** it utilizes the contextualized information for better paraphrase capturing than the EM. It computes the pairwise cosine similarity between the generated answer and the gold reference based on their generated BERT word representations to form an alignment used to produce a weighted F1 score based on inverse document frequency (idf) values [45].
- **Keyphrase Predictor for Question Answering (KPQA):** it is proposed specifically to evaluate the generative QAS based on Keyphrase prediction [46]. KPQA assigns different importance weights to each token in the generated answer, instead of treating them equally as done in the traditional n-gram-based models, to assess their significance in capturing the key meaning of the ground truth answer for the question. KPQA aims to identify the salient keywords or keyphrases critical to the answer's correctness and assign them higher values using a pre-trained BERT-based classifier. KPQA claimed to have a higher correlation with human judgments with respect to the other existing metrics. Moreover, KPQA can also be integrated into metrics like BLEU.

Different categories of QASs adopt different evaluation metrics to assist their performance. In general, the QASs with extractive answers use F1 score and EM, the retrieval and ranking-based QASs adopt MAP and MRR, the cloze-style and multiple-choice systems use accuracy, while the abstractive-based QASs use metrics such as BLEU, ROUGE, METEOR, and many more.

3 Deep learning models for QAS

This section provides a summary analysis of the main research works that used deep learning in their question answering models and groups them according to the main Deep Learning category that they belong to. Figure 9 provides the adopted classification of the QASs according to their deep learning category. In addition, Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16 provide the details of these models with respect to their adopted word embedding, used dataset, and evaluation metrics with some concluding remarks about these models.

3.1 Convolutional-based models

The introduction of several techniques and improvements to the Convolutional Neural Network (CNN) was proposed in [47] to learn the question and answer distributional representations. For example, the use of layer-wise supervision with multi-layer CNN and the use of CNN with augmentation and discontinuous convolution. They pointed out that adopting a shared, not separate, hidden layer structure in the CNN for the question and the corresponding answer increased the performance since the results of the convolution filters will be similar. Moreover, they stated that the higher the number of convolutional filters, the higher the level of abstraction obtained by the network, and thus, the accuracy would be improved. They also bonded the information obtained from the L2-norm and inner product to form a new similarity metric to apply to the learned features.

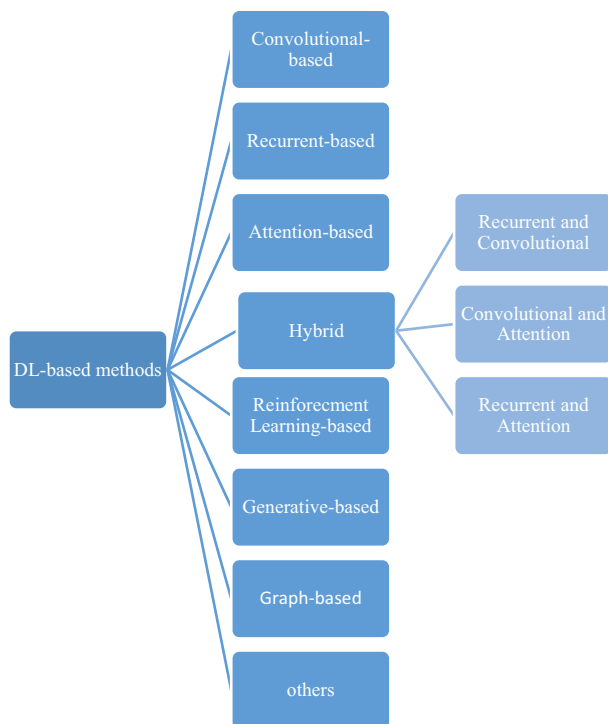


Fig. 9 Classification of QAS according to their used deep learning model

A Noise Contrastive Estimation (NCE) approach with a triplet ranking loss function was developed in [48]. The idea was to convert the answer selection task into a pairwise ranking task by exploiting “triplet” interactions to learn the relevance order of answer pairs with respect to the question. The triplet consisted of the question with positive and negative candidate answers. NCE was used for joint learning of the triplet representation to exploit, with the aid of the triplet ranking loss function, the nonlinear correlations to reduce the number of inversions required in the pair rankings. The NCE model consisted of two parallel pointwise base models, each associated with a pair of question and answer, to calculate the semantic similarity and generate representations. The model explored the performance of Multi-Perspective CNN [135] as their pointwise model.

A model that explored the role of the dissimilarities along with the similarities of the words in paired sentences was developed in [49] by decomposing and composing the words’ lexical semantics. Initially, each word in one sentence was represented by a low-dimensional distributed context vector to bridge the lexical gap problem between semantically equivalent sentences. After that, a semantic matching vector for each word in one sentence was calculated considering all words in the other sentence. Then, each word-matching vector was decomposed into similar and dissimilar components using different methods. The composition of all words with similar and dissimilar components and combining them into feature vectors was done with the help of a two-channel CNN model to capture multiple levels of granularity. Finally, a similarity score was given to the generated feature vectors in the answer sentence selection task for ranking purposes.

Table 2 Details of CNN-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[47]	2015	CNN	Word2Vec	InsuranceQA	The mixture of similarity metric, l2 norm, and inner product	Building and testing a domain-specific dataset.
[48]	2016	Rank MP-CNN: Multi-Perspective CNN with triplet ranking loss function	GLoVe	TERC QA tracks 8 to 13, WikiQA	MAP, MRR	Convert the answer selection as a ranking task instead of pointwise classification. Two different pointwise models were tested as base components at the word and sentence levels (+) Exploit existing pointwise models as plug-in components to achieve the model's flexibility
[49]	2016	L.D.C.: Lexical Decomposition and Composition	Word2Vec	QASent, WikiQA	MAP, MRR	(+) Three decomposition function types were tested: rigid that detected the exactly matched words; linear biased toward the similar component; and orthogonal decomposition that operated in the geometric space. The rigid performed the worst, while the orthogonal was the best.
[50]	2018	Question Classification	GLoVe	TERC QA tracks 8 to 13, WikiQA	MAP, MRR	(+) Concentrating on the Question Classification reduces the search space of potential answers by predicting the target class.

Table 3 Details of RNN-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[51]	2017	BiMPM: bilateral multi-perspective matching model	GLoVe	TERC QA tracks 8 to 13, WikiQA	MAP, MRR, Accuracy	Based on their similarities to the question, the answer sentence selection task aims to rank a list of candidate answers.
[52]	2017	HD-LSTM: Holographic Dual LSTM.	Word2Vec	TERC QA tracks 8 to 13, Yahoo's Web-scope L4 and nfl6	MAP, MRR	(+) An end-to-end holographic model with the ability of QA representations' scaling with minimal increase in the number of parameters through associative memory and circular correlation to learning the QA pairs relationship. (+) It reused knowledge from simpler supervised tasks to build the skill needed for the complex task.
[53]	2017	Skill model	GLoVe	SQuAD v1.1	EM, F1 score	

Table 3 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[54]	2020	Composition of question classification and LSTM network	GloVe	Heterogeneous dataset from TREC and USC datasets.	Accuracy	(+) Modular approach
[55]	2021	BNN: Bayesian Neural Network	GloVe	SimpleQuestions dataset using FreeBase KB with 2M entities (FB2M)	Accuracy, Area Under Receiver Operating Characteristic curve (AUC), Area Under Precision-Recall curve (AUPR)	(+) The model can obtain both the answer and its confidence; thus, it can be used easily (-) The model cannot handle the ambiguous questions

Table 4 Details of attention-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[56]	2018	knowledge distillation with RMR base model	GloVe	SQuAD v1.1, NarrativeQA	EM, F1 score	The main idea was ensemble model compression through a knowledge distillation framework that contains two networks: a teacher, a pre-trained large model, and a student, a smaller network that learns and is supervised by the teacher.
[57]	2019	OCN: Option Comparison Network	BERT	RACE	Accuracy	(+) They stated that this model was the first to surpass Amazon Mechanical Turker performance on the whole dataset.
[58]	2019	KT-NET: Knowledge and Text fusion NET	BERT	ReCoRD, SQuAD v1.1	EM, F1 score	(+) It improved the linguistic regularities of pre-trained LM by using additional knowledge from KBs (WordNet/ NELL).

Table 4 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[59]	2019	Megatron-BERT	BERT	Aggregated training dataset: Wikipedia, CC-Stories, RealNews, OpenWebtext//development: MNLI, QQP, SQuAD 1.1 and 2.0//Testing: WikiText103, LAMBADA, RACE	EM, F1 score, Accuracy	(+) Implementation of model parallelism to replace the traditional single-GPU-per-model for more efficient large-scale model training.
[60]	2020	TANDA: Transfer AND Adapt approach	BERT, RoBERTa	Training dataset: the constructed ASNQ dataset, Testing datasets: TERC QA tracks 8 to 13, WikiQA	MAP, MRR	(+) The intermediate fine-tuning step improved the stability of TANDA, while the adaptation step made it more robust to noise (+) They constructed the ASNQ dataset by converting the Natural Questions (NQ) corpus from machine reading to answer selection task
[61]	2022	DUMA: Dual Multi-head Co-Attention model	BERT	DREAM and RACE	Accuracy	DUMA layer was placed between the encoder and the decoder to simulate human transposition thinking process to capture the relationship between passage, question and answer options.

Table 4 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[62]	2020	REALM: REtrieval-Augmented LM	BERT	NaturalQuestions-Open, WebQuestions, CuratedTrec	EM, Accuracy	(+) Offer a set of model-centric unsupervised alignments between text in the pre-training corpus and knowledge corpus (-) Despite the cascaded retrieval module of S2G, no care about the sequential order of the retrieved paragraphs that mimic multi-hop nature are given since all of them are selected at once.
[63]	2021	S2G+ EGA: select-to-guide strategy with Evidence Guided Attention	BERT, RoBERTa, ELECTRA, ALBERT	HotpotQA dataset in the distractor setting	EM and F1 on answer, supporting facts, and the combination	
[64]	2022	FE2H: From Easy to Hard, a Two-stage Selector and Reader	ELECTRA, ALBERT	HotpotQA dataset in the distractor setting	EM and F1 on answer, supporting facts, and the combination	(+) FE2H reader progressively begins the training with the “easy” and “simple” SQuAD single-hop dataset before moving to the “hard” multi-hop dataset.
[65]	2022	Block-Skim: Transformer-based	BERT	SQuAD v1.1, Natural Questions, TriviaQA, NewsQA, SearchQA, HotpotQA	EM, F1 score	(+) Block-Skim is an add-on component that does not affect the backbone model but only regularizes its attention values and can be removed without affecting it. Block-Skim is proposed for extractive and multi-hop QAS.

Table 5 Details of hybrid convolutional and recurrent-based QAs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[8]	2017	A-ARC model: Combination of CNN, attention-based LSTM, and CRF.	Word2Vec	SemEval-2015 cQA	F1 score, Accuracy, Precision, Recall	(+) LSTM modeled the dependencies through learning correlations along the timeline of the QA sequence. Attention addressed information loss. In contrast, CRF captured label dependency in the comment quality labels.
[66]	2018	RCNN: combination of CNN and RNN.	Word2Vec	SemEval-2015 cQA	Precision, Recall, F1 score	(+) A two-phase learning strategy was designed in RCNN; modeling semantic relevance between question and answer and modeling semantic correlations in the answers sequence.
[67]	2018	Proposed many models: (1) two-layer LSTM. (2) CNN/TF. (3) LSTM/ CNN/TF. (4) Char+ Word/ CNN/ LSTM. (5) Memory/ CNN/LSTM/TF	Not stated	The constructed WikiPassageQA	MAP, MRR, Precision, Recall, Kappa coefficient	(+) Addressed the answer passage retrieval task and identified the beginning and end of the answer portion in the document (-)

Table 6 Details of hybrid convolutional and attention-based QAS

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[68]	2017	AI-NN: attentive interactive neural network.	GloVe	SemEval-2016 task 3 CQA	MAP, F1 score, Accuracy	(+) A 3D tensor was used to obtain the summarized interaction vector that contained the segments' importance and thus guided the attention.
[69]	2017	Dynamic-Clip Attention	GloVe	TERC QA tracks 8 to 13, WikiQA	MAP, MRR	(+) It used a list-wise training approach to learn the relative order of candidate answers and rank them.
[70]	2019	Compare Aggregate Model + LM +LC+TL	ELMo	TERC QA tracks 8 to 13, WikiQA, QNLI	MAP, MRR	(+) The sentence-level answer-selection task replaced the word embedding layer with a pre-trained LM to improve the words' contextual representations.

Table 6 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[71]	2018	CNN with attention mechanism and matching matrix.	Word2Vec	TERC QA tracks 8 to 13	MAP, MRR	(+) No complex feature engineering or additional statistical features were needed. The model used a complementary data augmentation technique called back-translation to obtain text paraphrases using two sequential translation models from English to French and vice versa
[72]	2018	QANet: the encoder consists of convolution and self-attention.	GloVe	SQuAD v1.1	EM, F1 score	(+) The recurrence-free model enhanced the training speed.

Table 7 Details of hybrid recurrent and attention factoid-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[73]	2016	KV-MemNN: Key-Value Memory Networks.	Pre-trained vectors using Supervised Embeddings	WikiQA, WIKIMOVIES	MAP, MRR, Accuracy	KV-MemNN is a generalization of Memory Networks. (+) They developed a QA dataset in the movie domain called WIKIMOVIES that contains raw text with a preprocessed KB
[74]	2016	aNMM: attention-based Neural Matching Model with value-shared weighting and matching matrix.	Word2Vec	TERC QA tracks 8 to 13	MAP, MRR	Ranking short answers task (+) No manual features or linguistic tools were needed. In position-shared weights, the node weight only depends on its position as defined by the CNN filters. On the other hand, in value-shared weights, the node weight depends on its value, i.e., the question and answer matching score.
[75]	2019	Composition of transformer-based with Bi-LSTM	Word2Vec	WikiQA	MAP, MRR, Accuracy	The model focused on sentence embedding (+) Bi-LSTM was used to incorporate the sequential features rather than employing position encoding.

Table 8 Details of memory-based recurrent and attention reading comprehension-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[76]	2015	MEMNN: Memory Neural Networks.	One-hot Vectors	bAbI	Perplexity	An end-to-end trainable model. They used Bag-of-Words (BoW) for sentence representation.
[77]	2016	DMN: Dynamic Memory Network.	GloVe	bAbI	Accuracy	(+) A trainable end-to-end model captured the position and temporality by employing a mixture of input representation, attention, and response mechanism sequences.
[78]	2017	MEMN: Multi-Layer Embedding with Memory Network	Word2Vec	TriviaQA, SQuAD v1.1	EM, F1 score	(+) The developed hierarchical attention vectors utilized the advantages of both one and two-dimensional attention with less required time and memory requirements.
[79]	2018	DEBS: Dense Encoder Block with Self-Attention	GloVe	TriviaQA, SQuAD v1.1, QUASAR-T	EM, F1 score	An advanced memory-augmented model consisted of three main layers: the co-attention, the memory controller, and the prediction layer.

Table 9 Details of co-attention-based recurrent and attention-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[80]	2016	DCN: Dynamic Co-attention Network.	GloVe	SQuAD v1.1	EM, F1 score	DCN is an end-to-end network with a novel co-attentive encoder and dynamic decoder.
[81]	2018	Co-matching Approach	Glove	RACE	Accuracy	(+) The model deal with the questions with evidence scattered in different sentences in the passage.
[82]	2017	DCN+: improved DCN with a deep residual co-attention encoder	GloVe	SQuAD v1.1	EM, F1 score	(+) The model had mixed objectives: reinforcement learning focused on textually similar answers to the ground truth answer, while the cross-entropy focused on correct answer trajectories.

Table 9 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[83]	2018	MQAN: Multitask QA Network with dual co-attention and multi-pointer-generator decoder.	GloVe	SQuAD v1.1 (for the QA task)	F1 score	(+) MQAN revealed TL and zero-shot capabilities after training on decaNLP, thanks to its multi-pointer-generator decoder.
[84]	2018	SLQA+: Semantic Learning for Question Answering	GloVe with ELMo	SQuAD v1.1, AddSent/AddOne Sent datasets, TriviaQA	EM, F1 score	(+) To mimic human reading pattern, the horizontal and vertical attention and fusions were conducted across layers to provide different granularity levels between questions and paragraphs.
[85]	2018	MCAN: Multi-Cast Attention Networks	GloVe	TERC QA tracks 8 to 13, Ubuntu Dialog Corpus (UDC), SemEval -2016 cQA, Tweet Reply Prediction	MAP, MRR, Precision, Recall	(+) The attention in MCAN was used as feature augmentation, i.e., casted attention. Also, three compression methods to convert the attentional matrices into scalar features were developed.

Table 10 Details of transfer learning-based recurrent and attention-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[86]	2017	TL technique	Initialized from a source QA dataset	WikiQA, SemEval-2016 cQA	MAP, MRR, Precision, Recall	A coarser, sentence-level QA through a standard TL technique of a model trained on a large, span-supervised QA dataset.
[87]	2017	SynNet: two-stage synthesis network	GloVe	SQuAD, NewsQA	F1 score	SynNet generalized a pre-trained model based on SQuAD dataset and fine-tuned on NewsQA dataset. Answers were generated before the questions because they held key semantic concepts.

An answer selection framework that integrated redefined fine-grained question classification taxonomy was introduced in [50], and a complementary entity identification system was created to match the new set of question classes. The answer selection model used with this newly modified classification was proposed in [48], and it highlighted entities that were generated using different word embedding methods.

Table 2 lists the specifications of convolutional-based QASs. As can be noted, these models are outdated and used by the earlier deep learning-based QASs since they have limited efficiency.

3.2 Recurrent-based models

Recurrent Neural Network (RNN) is the deep model that best reflects human thinking by building the results based on past sequences. RNN differs from the regular feed-forward neural network by having a feedback loop and small internal memory. Long Short-Term Memory Unit (LSTM) [136] is a variation of RNN proposed to tackle the long dependency and the vanishing gradient limitations of RNN. It converts the model into a memory cell using the concept of gates that control the cell states. A simplified version of LSTM called Gated Recurrent Unit (GRU) [5] couples two of its gates into a single one. All RNN variations, either LSTM or GRU, capture backward dependencies without exploiting the future context. Thus, in order to discover the context relations from the two directions: the future (forward

Table 11 Details of multi-hop based recurrent and attention reading comprehension-based QAAs

Ref No	Publishing year	The QA Deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[88]	2016	BiDAF: Bi-Directional Attention Flow network.	GloVe	CNN/Daily Mail, SQuAD v1.1	EM, F1 score	(+) Memory-less mechanism with multiple attention fusion from previous layers and different timesteps was used to reduce information loss due to early summarization.
[89]	2018	SAN: Stochastic Answer Network	GloVe	SQuAD v1.1, MS MARCO	EM, F1 score	Three additional linguistic features were used: POS, NER, and binary EM features. (+) Stochastic dropout in answer module improved robustness and accuracy.
[90]	2019	Weakly supervised bridge reasoner and passage reader	GloVe	full-wiki HotpotQA // “bridge” questions	EM, F1 score	The initial top 10 passages for each question were retrieved using a hybrid approach that combined tf-idf and bm25 score.

Table 12 Details of hybrid recurrent and attention reading comprehension-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[91]	2015	Deep LSTM Reader, Attentive Reader, Impatient Reader.	One hot encoding	CNN/Daily Mail	Accuracy	A semantic parser was used (+) Visualization of the inference process using heat maps was done.
[92]	2016	AS reader: Attention Sum Reader.	Using their own embedding function	CNN/Daily Mail, Children's Book Test (CBT) Common Nouns + Named Entities	Accuracy	(+) The ensemble of models yielded better results than using the single model with pre-trained word embedding.
[93]	2016	EpiReader.	Trainable embeddings	CNN, CBT	Accuracy	The task of the EpiReader was to answer a Cloze-style Question.
[94]	2017	Match-LSTM and Answer Pointer.	GloVe	SQuAD v1.1	EM, F1 score	An end-to-end model (+) They stated that the boundary model had outperformed the sequence model.
[95]	2017	AoA: Attention-over-Attention reader.	Randomly initialized with uniform distribution	CNN/CBT Named Entities and Common Nouns.	Accuracy	(+) AoA Reader utilized the mutual information in both the document and the query. Simple and general model.
[96]	2017	GA Reader: Gated-Attention Reader.	GloVe	CNN/Daily Mail, CBT's two subsets, Who Did What dataset	Accuracy	(+) GA provided a boost in the absence of feature engineering or increased training set size.

Table 12 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[97]	2017	Gated attention-based recurrent networks and Attention Mechanism.	GloVe	SQuAD v1.1	EM, F1 score	The end-to-end model consisted of 4 components: (1) a recurrent encoder to build separate questions and passages representation, (2) a gated matching layer to match the question and passage, (3) a self-matching layer to aggregate information from the whole passage, (4) a pointer network for answer boundary prediction.
[98]	2017	PhaseCond: phase conductor	GloVe	SQuAD v1.1	EM, F1 score	(+) PhaseCond is a multi-layered attention model that benefits from POS and NER tag features and EM binary features to determine if the examined passage word matches with any question word and vice versa.
[99]	2017	FusionNet	GloVe	SQuAD v1.1, AddSent, AddOneSent,	F1 score, EM	(+) “history of word” concept characterized attention information. FusionNet had: Word-level, High-level, and Self-boosted fusions.

Table 12 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[100]	2018	Hierarchical Attention Flow	GloVe	RACE	Accuracy	The attention at the word level: Question-to-Passage, and Question-to-Option; and at the sentence level, Option-to-Passage attention.
[101]	2018	AMANDA: question-focused multi-factor attention network	GloVe	NewsQA, TriviaQA, SearchQA	EM, F1 score	AMANDA used a tensor-based attention mechanism.
[102]	2018	S-Net: extraction-then-synthesis framework	GloVe	MS-MARCO dataset	ROUGE-L and BLEU-1.	The answer synthesis model took the extracted evidence as features fed to the sequence-to-sequence model.
[103]	2018	SDNet	BERT	CoQA	F1 score	(+) SDNet model comprehended dialogue flow and integrated it with passage content digestion (-)
[104]	2020	GF-Net: gated feature network	GloVe with ELMo	SQuAD v1.1	F1 score, EM	The gate-based feature effectiveness defined the answer boundaries in the encoding and pointing layers.

Table 13 Details of graph-based QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[105]	2019	KEQA: Knowledge Embedding-based Question Answering	GloVe, BERT	KG subsets from freebase FB2M, FB5M, SimpleQuestions dataset	Accuracy	QA over KG. (+) The designed joint distance metric considered KG embedding structures and relations; thus, it was able to handle ambiguity. (–) The target question type was simple questions that involved only a single head entity and a single predicate.
[106]	2020	GRAPHFLOW: Graph Neural Network	GloVe	CoQA and QuAC	F1 score	The nodes in the dynamic constructed context graph (in reasoning layer) modeled a passage word encoded (in encoding layer) a combination of the passage, the question, and the conversation history. The matching score between learned graph nodes and question embeddings were used to predict the answer (in prediction layer).

Table 13 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[107]	2022	CGDe-FGIn: Coarse-grain Decomposition Fine-grain interaction	GLoVe	SQuAD and HotpotQA datasets	EM, F1 score	The model avoided using expensive grammatical tools such as NER for the graph construction and used lightweight models instead.
[108]	2019	QFE: Query-Focused Extractor	Word2Vec	HotpotQA	EM, F1 score	QFE was also evaluated on the textual entailment FEVER dataset.
[109]	2019	DFGN: Dynamically Fused Graph Network	BERT	HotpotQA in the distractor setting	EM, F1 score	They proposed a metric for evaluating the quality of the generated reasoning chains. The questions and passages were tokenized using an uncased BERT Tokenizer.
[110]	2019	BAG: Bidirectional Attention entity Graph convolutional network	ELMo, GloVe	QAngaroo, WIKIHOP	Accuracy	It combined two representations: Glove to provide tokens to representations and ELMo to provide contextual node representations to account for the document and their positions in the graph.

Table 13 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[111]	2020	HGN: Hierarchical Graph Network	BERT, RoBERTa	HotpotQA	EM, F1 score	Title matching was used to select the relevant paragraphs. HGN has 7 edges between: question node and (paragraph nodes and its entity nodes), paragraph nodes and their corresponding sentence nodes, sentence nodes and their (linked paragraph nodes, corresponding entity nodes and neighbor sentence nodes within the same paragraph), and paragraph nodes.
[112]	2020	CFGNN: Coarse and Fine Granularity Graph Network	BERT	HotpotQA dataset in the distractor setting	EM and F1 on answer, supporting facts, and the combination	CFGNN was inspired by using a multi-granularity graph to manipulate the context introduced in DFGN [109].

Table 13 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[113]	2020	SAE: Select, Answer and Explain system	BERT and Roberta	HotpotQA	EM and F1 on answer, supporting facts, and the combination	The questions and passages were tokenized using the wordpiece tokenizer. The named entities and noun phrases in context and question were recognized using spaCy3.
[114]	2020	Transformer-XH: transformer with eXtra Hop	BERT	HotpotQA	EM and F1 on answer, supporting facts, and the combination	BERT ranker was used for information retrieval. The final model uses three-hop steps. Transformer-XH also operated on the FEVER dataset that aim of verification label classification against textual sources
[115]	2022	GREASELM: Graph REASONing Enhanced LM	RoBERTa-Large	Commonsense QA, OpenbookQA	Accuracy	The used KG is Concept Net and then it is evaluated using multiple-choice QA datasets

Table 13 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[116]	2021	TransferNet	BERT	MetaQA, WebQSP, CompWebQ	Accuracy	TransferNet performed qualitative analysis by visualizing intermediate results of its attention for answer interpretability and transparency.
[117]	2021	IDRQA: Iterative Document Reranking phase and a QA phase	BERT, ALBERT	Natural Questions Open, SQuAD Open, HotpotQA.	Open, EM and F1 on answer, supporting facts, and the combination	Contextual Encoding was done by feeding each document independently along with the question to the pre-trained LMs. It targeted the open-domain questions
[118]	2021	LEGO: Latent Execution Guided reasoning framework	BERT	MetaQA, WebQSP, CompWebQ	accuracy using Hits@1 metrics	execution-guided search space pruner was used to limit the search space.
[119]	2021	MDR: Multi-hop Dense Retriever	RoBERTa	HotpotQA	Precision, recall, F1 score	No corpus-specific information or graph structure, such as inter-document hyperlinks or human-annotated entity markers, was needed. MDR was tested on the FEVER dataset as well.

Table 14 Details of generative-based QAs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[120]	2020	CABIN: cross-sentence Context Aware Bidirectional LSTM	GLoVe	TREC, Yahoo! StackEx(L), WikiQA	MRR, MAP, mean ranking of the top-N answers	In the text preprocessing, special symbols to indicate the end of sentence and OOV words were used.
[121]	2019	KEAG: Knowledge-Enriched Answer Generator	GloVe	MARCO	BLEU-1, ROUGE _L , human evaluation on Amazon Mechanical Turk	ConceptNet: a semantic network that combines words, phrases, and their commonsense relationships as external knowledge.
[122]	2018	MHPGM: Multi-Hop Pointer-Generator Model	ELMo	generative NarrativeQA// Generalizability testing: QAngaroo-WikiHop extractive dataset.	BLEU-1, BLEU-4, METEOR, Rouge-L// WikiHop dataset: Accuracy	They assisted the quality of the selected commonsense relations and the answers using conduct human evaluation to check the degree of agreement.

Table 14 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[123]	2021	Sequence-to-sequence two-step based model	BERT	TriviaQA, NaturalQuestions, SQuAD v1.1	EM, F1 score	SpaCy was used for tokenization. Text normalization was performed by lowercasing and removing (articles, punctuation, and duplicated whitespace). The retrieval methods are DPR (dense representation) for TriviaQA and BM25 for SQuAD.
[124]	2022	PATHFID: multi-hop reasoning generative approach	T5-large	HotpotQA, IIRC dataset	EM and F1 on answer, supporting facts, and the combination	The reasoning path: the question → first paragraph → second paragraph PATHFID generative reader operates for K-hop and jointly generates an alternating sequence of passage-level and fact-level

Table 15 Details of reinforcement learning-based QASs

Ref No	Publishing year	The QA Deep Learning-based Model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[16]	2017	Parallelizable CNN and reinforcement learning with hard and soft attention.	One hot vectors	WIKIREADING, WIKIREADING-LONG, WIKISUGGEST	Accuracy	The coarse-to-fine hierarchy was inspired by human ways of initially skimming the document, then identifying relevant parts to read them carefully to find the answer.
[125]	2017	ReasoNet: Reasoning Network trained with reinforcement learning	GloVe	CNN/Daily Mail, SQuAD v1.1, structured Graph Reachability dataset.	EM, F1 score	ReasoNets utilized the reinforcement learning instance-dependent reward baseline for training. They also developed the structured Graph Reachability dataset.
[126]	2018	Active Question Answering Model with reinforcement learning.	GloVe	Pre-training dataset: the multilingual United Nations Parallel Corpus v1.0. Evaluation Dataset: SearchQA	F1 score, EM	The aim was to provide QAS with multiple natural language reformulations of asked question and combine the returned evidence to construct best quality answer using an end-to-end model and policy gradient.

Table 15 (continued)

Ref No	Publishing year	The QA Deep Learning-based Model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[127]	2017	DFN: Dynamic Fusion Network	GloVe	RACE	Accuracy	The on-the-fly chosen attention strategy for each sample gave the DFN model more flexibility toward different question types.
[128]	2018	RMR: Reinforced Mnemonic Reader	GloVe with ELMo	SQuAD v1.1, AddSent, AddOneSent.	EM, F1 score	Avoid the problems of attention redundancy by utilizing iterative multi-round alignment architecture with a reattention mechanism. RMR replaced the traditional static reward and baseline with dynamic ones to address the convergence suppression problem.

Table 16 Details of other QASs

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[129]	2017	RNs: Relation Networks	Using LSTM	bAbI	Accuracy	RNs were used as a plug-and-play module to solve relational reasoning-based problems.
[130]	2017	Cascaded Approach of feed-forward networks with an attention	GloVe	TriviaQA	EM, F1 score	(+) They stated the ability of the model to scale large evidence documents. The separation into submodels and the multi-loss objective prevents the adaptation issue between features. Also, the non-recurrence nature enabled the processing of longer evidence texts consisting of simple submodels, which provide model scalability (−) Despite its question-in-span feature, entity-type confusion exists among the different levels.

Table 16 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[131]	2018	HyperQA: is a parameter-efficient neural network.	GloVe	TERC QA tracks 8 to 13, WikiQA, SemEval-2016 cQA, and YahooCQA	MAP, MRR, Precision	(+) HyperQA is a model with no feature engineering, no similarity matrix matching, no complicated attention mechanisms, and no over-parameterized layers. The Hyperbolic space granted the model self-organizing ability and automatic discovery of hierarchies (−) Their projection layer used Riemannian-SGD to learn the embeddings, which increased model complexity.
[132]	2020	BLANC: BLock Attention for Context Prediction	BERT	SQuAD, NaturalQ, NewsQA, HotpotQA	EM, F1, Span-EM (EM of answer span), span-F1	(+) BLANC also predicted the supporting facts in HotpotQA using zero-shot learning using a model trained on SQuAD.

Table 16 (continued)

Ref No	Publishing year	The QA deep learning-based model	Used word embedding technique	Used QA dataset	Used evaluation metric	Comments
[133]	2021	CAGKG: Community Answer Generation method based on the K G	Phrase embedding with a three-layer Skip-Phrase model	Stack Overflow for computer programming, Super User for computer enthusiasts, mathematics for math-related questions, and Quora.	BLEU, ROUGE, PASSE	More preprocessing work was required in CAGKG, such as phrase mining and phrase embedding. A new Semantic Similarity Evaluation indicator, PASSE, was proposed based on phrases. However, this similar metric is more complicated to evaluate than the word-matching ones.
[134]	2021	DFM: Deep Fused Model	Word2Vec	Simple-Questions and FB2M and FB5M), Chinese Dataset	Accuracy, average f1, lexical-level similarity	KB Question Answering (+) A parameter-shared DFM integrated two complementary sub-tasks and trained them jointly to enhance the semantic representation (-) The elimination process of noisy and ambiguous candidates can harm the recall rate. Also, the complete contextual information cannot be captured by only utilizing the matching subject and predicate information between the question and KB

dependency) and the past (backward dependency), the bidirectional RNNs are proposed in [137] that achieve this with the help of two separate hidden layers. The outputs of these two hidden layers are aggregated to generate the output of the current time step.

The Bilateral Multi-Perspective Matching (BiMPM) model [51] was a matching-aggregation model that consisted of five layers. The first layer was a word representation layer that converted the question and passage sentences into vectors using both character and word embeddings. Then, a bidirectional LSTM (Bi-LSTM) encoder was utilized in the context representation layer to encode and integrate the needed contextual information in the obtained representation vectors. The matching layer matched the two encoded sentences, the question, and the passage, in two directions, where a sentence in each time step was matched against all time steps of the other sentence in each direction. Afterward, these matching vectors were combined into a fixed-length vector using another Bi-LSTM in the aggregation layer. Lastly, the final prediction layer was made from a two-layer feed-forward network with softmax activation.

The Holographic Dual LSTM (HD-LSTM) [52] modeled the correlation between question and answer representations and re-ranked the question–answer pairs. This was done using a set of two multiple layers LSTMs (Question-LSTM and Answer-LSTM), and then, their outputs were fed to the holographic composition to match this pair. This provided a low-cost scalable, rich representation.

The ‘skill’ model [53] transferred and injected the obtained knowledge from other NLP tasks, such as textual entailment, NER, paraphrase detection, and question-type classification tasks, into the reading comprehension model. This was done through two steps: the Skill Learning step, where the encoder was trained for different tasks other than question answering, and the Neural Skill Transfer step, where the weights of the encoder in the first step were used to encode the passages and questions words in the QAS.

A combination of the knowledge of the question classification with an LSTM network was suggested by [54]. This architecture was divided into two models: the basic model for the questions’ main class classification and the second for the subclass classification. This created a dependency between the question and the primary classification to associate the question subcategory and thus contextualize the answer.

The end-to-end Bayesian Neural Network (BNN) model [55] used Knowledge Bases. The framework simultaneously selected the entity and relevant predicate from the KB to be encoded using a Bayesian Bi-LSTM to avoid uncertainty propagation. Two types of uncertainties were estimated: model uncertainty, which indicated how well the model fits the data, and predicted data uncertainty, which indicated how much inherent noise exists in the data.

Table 3 lists the specifications of recurrent-based QAS. As can be noted, these models are usually combined with static word embedding.

3.3 Attention-only-based models

Attention mechanisms made their way into various NLP tasks, trying to mimic human visual attention, including the QASSs. The attention mechanism’s powerfulness is represented by its ability to allow the model to focus on the essential and relevant features in different input positions. This is achieved by taking a weighted arithmetic mean of the inputs fed to the network, where each weight is assigned based on the relevance of the corresponding input to this given context. Different attention mechanisms differ in how these weights are assigned and how the weights average is derived according to the context.

A knowledge distillation-based framework was developed in [56] to benefit from the power of ensemble model training in reading comprehension tasks and reduce their expensive cost. This transferred the knowledge from an ensemble model to a single model with comparable performances. The framework consisted of joint training of a vanilla knowledge distillation, a developed answer distillation, and a developed attention distillation. The vanilla distillation transferred knowledge to learn the relative similarities of answer positions, but the model was biased and over-confident toward the confusing answers. Thus, the answer distillation addressed that by penalizing the most confusing answer span with a margin loss. On the other hand, the attention distillation guided the training using attentive information distilled from the ensemble model to capture precise question and passage interactions. In short, a new dataset was constructed using the different knowledge types distilled from the trained ensemble model, and then, a new single model was trained using that constructed dataset. The RMR [128] was used as the base model.

The Option Comparison Network (OCN) [57] aimed to enhance the reasoning ability and to mimic the human strategy in solving reading comprehension MCQ; having a general overview of the article, then studying the options in the question to build the required correlation and afterward a detailed reading of the article with respect to these correlations. Using OCN, each option was vectorized using a skimmer network. The correlation between each and every option vector was explicitly obtained at the word level. Finally, the article was re-examined to compute the probabilities for each option to be the candidate answer using self- and co-attention mechanisms.

The Knowledge and Text fusion NET (KT-NET) [62] deployed different integrated techniques to increase the prediction's effectiveness for machine reading comprehension tasks. The attention mechanisms were used to adaptively highlight the relevant information from different KBs, such as WordNet [20], and BERT contextual embedding. After processing the question and the passages, the potentially retrieved KB embeddings were stored in the knowledge memory. Afterward, the BERT encoded representations were integrated with the selected KB from memory, and then, the rich interaction between them was captured after the fusion using a self-matching layer. Finally, a knowledge-aware prediction was made at the output layer.

A simple intra-layer parallel approach, called Megatron, was proposed in [59] that allowed the training of huge transformer-based models with billions of parameters. Transformers [34] are networks that capture words' mutual influence by applying a multi-head self-attention mechanism. The powerfulness of Megatron lies in replacing the layer normalization in BERT-like models with attention.

A two-step fine-tuning approach called Transfer AND Adapt (TANDA) was proposed in [60] for training the transformer-based models [34], which are sequence transduction models that are based entirely on the attention mechanisms without relying on any recurrences or convolutions. The first step in TANDA was concerned with transferring the transformer model. In contrast, the second fine-tuning step adapted the obtained general model to the certain target domain with its specific question and answer nature. This approach helped take advantage of the pre-trained powerful models on a large high-quality dataset and adapt the obtained knowledge to the target domain, making the fine-tuned model more stable and robust to noise. Similarly, a Dual Multi-head Co-Attention (DUMA) model [61] captured passage-question relationships for multiple-choice machine reading comprehension tasks.

The different LMs usually implicitly store the obtained knowledge representations in their parameter weights, implying the need for more complex larger networks for better understanding. REALM, Retrieval-Augmented LM, was proposed in [62] to explicitly capture

modular and interpretable knowledge required in the open QAS by following a retrieve-then-predict approach. The REALM architecture consisted of two components: neural latent knowledge retriever and Knowledge Augmented Encoder. The first augmented the LM by retrieving the relevant documents needed to answer the question from a large textual knowledge corpus. The encoder utilized the question and these retrieved documents to predict the answer. The knowledge retriever was trained with the salient span masking cloze-style unsupervised training objective in an end-to-end manner; it used BERT-like vector embeddings fed to the transformer model.

A coarse-to-fine evidence retrieval strategy for multi-hop reasoning called Select-to-Guide (S2G) was proposed in [63]. S2G had a Multi-Head Self-Attention layer that incorporated two novel attention mechanisms; Sentence-aware Self-Attention (SaSA) and Evidence Guided Attention (EGA). The former explicitly aggregated all the sentence token embeddings, while the latter directed the model's focus toward the extracted evidence. The two main components of S2G are the cascaded paragraph retrieval and the multi-task modules. The first consisted of a score matching submodule, a score refinement submodule, and a fine-grained evidence paragraph retrieval submodule that operated according to multi-hop dependency and inter-paragraph cross attentions. On the other hand, the second module simultaneously extracted evidence sentences and answer spans and consisted of one shared encoder and two inter-dependent modules for each attention mechanism.

A simple multi-hop reasoning framework called From Easy to Hard (FE2H) was proposed in [64] with a two-stage document selector and reader that mimics the human's decomposition and progressive learning. The first stage in the FE2H selector identified the most relevant document to the question. In contrast, the second stage considered iteratively and accumulatively the relevancy based on both the question and the selected document to find other related documents. The selector aimed to filter the information and provide the reader with a noise-free, high-quality context. The first stage in the FE2H reader was a single-hop QA module that was then transferred into a multi-hop module with a linear prediction layer in the second stage to provide answers and their supporting facts jointly.

Block-Skim, a plug-and-play module in transformer-based backbone model, proposed in [65] that filters the information in the context paragraph as relevant to the process or unnecessary to discard according to the question in order to accelerate the answer prediction. At first, a block relevance predictor based on the self-attention weights of the transformer model is adopted and operated on the context blocks to make a skim decision. Then, the hidden states in the lower layers are further pruned by jointly training an end-to-end multi-objective single-task.

Table 4 lists the specifications of QASs that are only dependent on attention. Most of these models used the recent pre-trained contextualized embedding to encode the words.

3.4 Hybrid models

3.4.1 Convolutional and recurrent-based models

The Attentive Deep Neural Network ARChitecture (A-ARC) [8] took advantage of the collaboration of CNN, LSTM-based attention mechanism, and Conditional Random Fields (CRF). This integration resulted in more flexibility in the input format and the model's internal substructures to learn the deterministic information for the answer selection task.

Another integration of RNN and CNN was reformulated in the Recurrent Convolutional Neural Network (RCNN) model [66] to explore the semantic matching between the question

and the corresponding answer. Also, it captured the semantic correlation in a sequence of potential answers for answer selection in the Community QAS. The capabilities of various neural architectures and retrieval models to retrieve a passage that answers the question were explored in [67], specifically, identifying where the answer begins and ends within the retrieved passage. They conducted the experiments using their novel proposed non-factoid answer passage retrieval corpus called WikiPassageQA; more dataset description is given in Sect. 4.

Table 5 lists the specifications of QASs that are based on the hybridization of recurrent and convolutional techniques; the reviewed models are community-based ones.

3.4.2 Convolutional and attention-based models

The Attentive Interactive Neural Network (AI-NN) was proposed in [68] for answer selection in Community Question Answering. A CNN was initially used to learn the question and the answer representations fed to the AI-NN network. The network collected the interactions or the matching patterns between each paired representation of question and answer through row-wise and column-wise pooling. Then, an attention mechanism was used to focus on the important segments in the text.

The attention mechanism named Dynamic-Clip Attention [69] was injected into the Compare-Aggregate model [138]. Dynamic-Clip Attention aimed to filter out the noise, i.e., to clip the attention score of the irrelevant words to zero, in order to exploit the semantic relevance between the question and the answer. They also replaced the pointwise classification approach of the classical Compare-Aggregate framework with a list-wise ranking approach to learn the question and the relative order of candidate answers jointly. Another enhancement of the compare-aggregate model was proposed in [70] by adopting a Latent Clustering (LC) method, pre-trained LM, and Transfer Learning (TL), and changing the list-wise objective function to point-wise. The LM provided the contextual representations of the words in the sentence to capture their semantic relations. At the same time, LC grouped question-answer pairs into similar clusters by comparing the similarity between the sample and the content of a latent memory to provide auxiliary information that improves the model.

An attention-based CNN model based on a matching matrix was proposed in [71] to extract the semantic information between the independent words of the question and its answer. Therefore, the word and phrase relationships at different granularities were used to compute the correct probabilities to re-rank candidate answers.

The QANet [72], a feed-forward model, consisted of only convolutions and self-attention. The local interactions were modeled using convolutions by capturing the text's local structure, while the self-attention modeled the global interactions between each pair of words. QANet consisted of five layers: an embedding layer, a context query layer, an attention layer, an encoder layer, and an output layer. Throughout the QANet model, the several encoders' blocks only differed by the number of convolutional layers. Each encoding block contained convolutional layer norm, self-attention, feed-forward layers, residual connections, and positional encoding.

Table 6 lists the specifications of QASs that depend on attention and convolution; most of these models are community-based QAS.

3.4.3 Recurrent and attention-based models

The Key-Value Memory Network (KV-MemNN) was introduced in [73]. A key-value structured memory was used to store the facts. Then, the model was learned to address the facts

from this memory that were most relevant to the question using keys and then, returned the values of the answer. The advantages offered by this model were its effectiveness and flexibility.

The attention-based Neural Matching Model (aNMM) [74] replaced the position-shared weighting scheme with a value-shared weighting scheme, where the value measured the similarity between two words. It relied on the fact that semantic similarity was more beneficial than spatial regularities when a matching relationship between the question and corresponding answer was needed; this was achieved using attention mechanisms.

A Transformer-based neural network for answer selection was proposed in [75]. The transformer was followed directly by a Bi-LSTM to obtain the global question and answer information, in addition to their sequential features. The model consisted of the word representation layer, the transformer-based feature extractor, and the relevance matching layer. They proposed three transformer-based models according to the chosen aggregation strategies in the relevance matching layer to generate sentence embeddings: Transformer-based model with weighted mean pooling (QA-TFWP), max-pooling (QA-TFMP), and attentive pooling (QA-TFAP).

Table 7 lists the specifications of QAs dependent on attention and recurrent models; most of these models are of type retrieval and ranking systems with factoid answers.

A powerful network architecture was proposed by [76] called memory networks. They added a long-term memory component to grant the inference components the ability to reason. This long-term memory was considered a dynamic KB and was used to enhance the prediction by reading from and writing to it. The Dynamic Memory Network (DMN) [77] was an enhancement of [76] by introducing episodic memory. The DMN is an architecture that consisted of multiple stages: constructing the context and the questions' representations, forming episodic memories, and generating relevant answers. An iterative attention process was used and triggered by the question; it iterated over the input and over the result of previous iterations, which consequently updated the memory with the newly relevant information about the input. Afterward, the answer was generated with the aid of a hierarchical recurrent model. The main advantages of this model were the modularity and transitive reasoning ability; it consisted of four main modules: an input module, a question module, an episodic memory module, and an answer module.

The end-to-end Multi-Layer Embedding with Memory Network (MEMEN) for the machine reading comprehension task was proposed in [78]. The passage's words and questions were encoded to combine the syntactic and semantic information in their embeddings using the SG model to train the Part-Of-Speech (POS) and NER tags. MEMEN had a memory network with full-orientation matching of the query and passage to catch more interactional information by combining the results of hierarchical multi-hop attention vectors. Then, a pointer network [139] was used to predict the answer boundaries.

The Dense Encoder Block with Self-Attention (DEBS), along with a modified memory controller with block and layer-wise residual connections, was proposed in [79]. These modifications aimed to improve the memory-handling capability of the system. The dense connections conveyed a rich data representation across multiple layers by concatenating a particular layer's input and output. This alleviated the potential information distortion and loss as the network got deeper. The self-attention was used to maintain the information across different time steps to tackle the long-term dependency issue of reasoning in a lengthy document.

Table 8 lists the specifications of QAs dependent on attention and recurrent models; these models are of type reading comprehension.

The Dynamic Co-attention Network (DCN) [80] consisted of a co-attention encoder that captured the interactions. Thus, constructing co-dependent representations helped concentrate on the relevant information by using a dynamic decoder that iteratively updated the answer span. This enabled the model to jump from the initial local maxima caused by incorrect answer predictions.

The co-matching approach for the multiple-choice reading comprehension task in [81] aimed to match the question–answer pair in the passage by scanning the passage and computing two attention-weighted vectors each time; matching the question and matching the candidate answer, each with a selected portion of the passage to form a co-matching state. After completing the passage, the generated sequence of co-matching states was fed to a hierarchical LSTM to aggregate the information from word level up to document level.

A deep residual co-attention encoder enhanced the DCN architecture [80] and formed DCN + [82]. This allowed each input to attend to the previous attention contexts and consequently build a richer representation. Also, they proposed a mixed objective that combined the traditional cross-entropy loss over positions with self-critical reinforcement policy learning deployed through a reward derived by word overlap between the predicted answer and the ground truth answer to solve the misalignment between the evaluation metric and optimization objective.

The Multitask Question Answering Network (MQAN) [83] framed all the NLP tasks in the decaNLP Decathlon challenge as a QA and dealt with them without any task-specific modules or parameters. It integrated many techniques, such as sequence-to-sequence learning, a multi-pointer hierarchical decoder based on pointer networks [139], and an anti-curriculum training strategy with dual co-attention.

A Hierarchical Attention Network was proposed in [84] to answer reading comprehension questions. Using co-attention and self-attention mechanisms, the fusion of different word representations, and the matching of a pointer network [139], the correct answer span in the passage was gradually defined by the continuous refining of the relationship between the question and passage at different levels of granularity.

The Multi-Cast Attention Networks (MCAN) [85] used casted attention for generic sequence pair modeling feature augmentation. MCAN performed a series of attention types, such as co-attention or self-attention, with different variants such as alignment-pooling, max-pooling, or mean-pooling to cast a scalar feature at each time step and re-attached it to the inner word embeddings. MCAN treated the attention outputs as features. These attention and compression operations allowed re-weighting of the word representation by taking into account both the global and the cross-sentence knowledge to improve the learning capabilities of the network. Then, the answer was provided by feeding the resulting enhanced embedding to a sequential encoder layer.

Table 9 lists the specifications of QASs dependent on attention and recurrent models; these models are reading comprehension with a specific type of reasoning called co-attention.

The famous TL concept of deep learning was first applied to the QA field in [86]; it investigated the ability to generalize a system trained using a specific dataset by applying another. They performed a basic transfer learning from a BiDAF model [88] trained on the SQuAD dataset to be used with target datasets: WikiQA and SemEval-2016 (Task 3A). Although these two datasets were characteristically different from the source dataset, this TL achieved good and comparable results.

The two-stage SYNthesis Network (SynNet) [87] used TL to answer questions in a domain where no labeled annotated data may be available, i.e., it generalized the target domain based on another source domain by fine-tuning the trained model. The synthetic resultant

question–answer pair was constructed by generating the answer conditioned based on the paragraph and generating the question by conditioning on both the paragraph and answer.

Table 10 lists the specifications of QASs dependent on attention and recurrent models. However, these models adopt the transfer learning concept in their model construction.

An extension of BiDAF [88] called Ruminating Reader was proposed in [140] that used a multi-hop attention mechanism to enhance the reasoning over the full context and avoid mistakes in answer span extraction. They also proposed a novel layer structure called a ruminate layer. It used gating mechanisms to generate a query-aware context vector representation and fuse the encoding representation obtained from the first and second passes of attention.

Another model that adopted multi-step reasoning for the reading comprehension task was introduced in [89] called Stochastic Answer Network (SAN). It was named after the stochastic prediction dropout applied to the answer module. The SAN model iteratively refined its prediction over a fixed number of reasoning steps. Then, it generated the answers based on the average of all these steps' predictions rather than solely on the final step. SAN consisted of a lexicon encoding layer that performed word embedding mapping; a context encoding layer that used Bi-LSTM to obtain context representations; an attention layer to derive a question-aware passage representation; and a self-attention layer to re-arrange the gathered information. Another LSTM was used as a working memory to serve the passage on the top of these layers. Finally, a GRU-based answer module outputted the predictions at each state.

A weakly supervised model called bridge reasoner was proposed in [90] to provide reasoning for multi-hop questions by following a context-aware passage retrieval. It retrieved the candidate answer passages that provided links to the question by utilizing multiple evidence types. The generated candidate passages were then fed to the passage reader module for answer span extraction with the help of an auxiliary objective to utilize the answer passage supervision. Both the bridge reasoner and passage reader shared the same GRU-based attention architecture.

Table 11 lists the specifications of QASs dependent on attention and recurrent models; these models involve multi-step or multi-hop reasoning to extract the answers.

The attention mechanisms were integrated with LSTM to overcome the problem of insufficient expressive power in [91] by introducing three models; the Attentive Reader, the Impatient Reader, and the deep LSTM Reader. These models were able to carry and accommodate semantic information over long distances.

The Attention Sum (AS) reader was proposed in [92], which was suitable for a single-word answer system. It consisted of two bidirectional GRU networks whose hidden states constructed the word embedding that captured the context from the entire document and, thus, provided more flexibility. However, this model failed to generate an answer that was not mentioned in the document.

The EpiReader model [93] consisted of two network components that worked sequentially; one network called Extractor selected a set of potential answers, while the other network called Reasoner reprocessed this set, then based on the question and the context, and the candidate answers were re-ranked. The Extractor network was a pointer network [139] that used a pair of bidirectional GRUs and a differentiable attention mechanism to indicate whether a potential word in the context was a candidate to be an answer.

The combination of the match-LSTM model and pointer network in [94] targeted the reading comprehension task. The match-LSTM model was used for predicting the textual entailment by matching the attention-weighted premise to each token of the hypothesis, then making the final prediction based on this aggregated matching result. On the other hand, the Pointer Network used an attention mechanism as a pointer on the input text token to extract

the answer. Two models were proposed: the sequence model and the boundary model. The pointer network in the former model ignored the consecutivity in the sequence tokens of the answer in the input passage. In contrast, in the latter model, the start and end tokens of the answer were selected from the passage.

The Attention-over-Attention (AoA) reader [95] introduced an additional attention mechanism to automatically focus on the important primary attentions obtained so far, thus introducing an “attended attention” that took advantage of the interactive information between the query and the document.

The attention mechanism in Gated-Attention (GA) Reader [96] was implemented using multiplicative interactions between the embedding of the question with the intermediate states of an RNN, i.e., the contextual embedding. This gated attention acted as a refined filter during the iterative hops over the input (the context) to enhance the reasoning.

The self-matching attention mechanism in the GRU in [97] was utilized to refine and shift the focus to the significant words throughout passages to their surrounding context windows. The model was aided using a pointer network [139] to specify the answer’s position and boundaries. The question and the passage representations were built separately using a bidirectional recurrent network and then, matched together in the gated matching layer.

The Bi-Directional Attention Flow (BiDAF) network [88] had a multi-layer structure that consisted of six layers, each capturing the context at different levels of granularity. BiDAF introduced three levels or layers of embeddings: character-level, word-level, and contextual embedding. The fourth layer was a bidirectional memory-less attention layer that considered two directions to get a query-aware context representation: query-to-context direction and context-to-query direction. This was followed by a modeling layer that scanned the context using RNN. Finally, the output layer produced the answer for that query corresponding to a span from the examined passage. Thus, a bidirectional attention matrix was constructed by BiDAF.

As its name suggested, Phase Conductor (PhaseCond) [98] consisted of two sequential phases where each phase contained two-layer stacks; the first (question-aware passage representation phase) consisted of multiple attention and outer fusion layers responsible for producing question-aware passage representations, while the second (evidence propagation phase) consisted of self-attention and inner fusion layers responsible for regulating and propagating the concatenated information flow. An improved attention mechanism for PhaseCond was also proposed that implemented two types of encoders: an independent question encoder and a weight-sharing encoder to jointly encode the question and the passage.

The FusionNet [99] introduced the “history of word” concept to extend the information used by the attention to consider all the information obtained all the way, starting from the lowest level word embedding to the highest level semantic representation. Moreover, to extract all the benefits and utilize this novel concept efficiently, they combined it with an attention score function. FusionNet exploited the obtained information layer by layer in a discriminating order.

The hierarchical attention flow in [100] handled the multiple-choice reading comprehension task by considering the effects of options correlations explicitly on boosting the performance of the model; i.e., it adopted different levels of attention to capture more sufficient relevant information from the passage with the help of both the question and the candidate answer. Then, each option of the question was evaluated with a score value using sentence-level attention and a fixed-length encoding and considering the other options.

The end-to-end question-focused multi-factor attention network for document-based QA (AMANDA) was introduced [101]. Using the max-attentional question aggregation mechanism, it learned to aggregate fine-grained meaningful, relevant information distributed across

multiple sentences in the passage. Also, it focused on the important words in a question to identify the question type and consequently the suitable corresponding answer type. Thus, a deeper understanding was provided, especially in the long context in which a multi-sentence reasoning or co-reference resolution may be required.

The extraction-then-synthesis framework (S-Net) [102] was based on sequence-to-sequence modeling. As the name implies, the most relevant passage sub-spans to the question are extracted and considered as pieces of evidence fed as additional features together with the question and the passage to the answer synthesis model to produce the final answer.

The SDNet [103], a contextualized attention-based deep neural network, addressed the conversational QA task by utilizing both inter-attention and self-attention techniques on the passage and the dialogue question history for incorporating more effective contextual understanding. SDNet broke the canonical way of dealing with BERT [27] contextual embedding; i.e., instead of fine-tuning according to the dataset by adding an extra layer, they fixed the BERT model parameters and took linear weighted combinations of the output of different BERT layers.

The Gated Feature network (GF-Net) [104] depended on the linguistic features for the machine reading comprehension task. These linguistic features were chosen automatically through feature gate mechanisms according to their participation and effectiveness in the answer selection process. The GF-Net architecture was decomposed into three layers: an encoding layer where static, dynamic, and character-based word embeddings were used for encoding the question and context; the interaction layer where a combination of bidirectional attention and self-attention mechanisms were used to define the relations and interactions between the question and the context; and finally, a pointing layer where a pointer network [139] with feature gates was used to identify the answer starting and ending positions.

Table 12 lists the specifications of QASs dependent on attention and recurrent models. Most of these models were of type reading comprehension and used static word embedding techniques.

3.5 Graph-based models

A graph neural-based model, called GRAPHFLOW, was proposed in [106] for the conversational machine reading comprehension task. It captured the global temporal dependencies among context words by dynamically constructing a context graph enlightened by the question from the passage text at each turn. This sequence of graphs was processed with the previous turn reasoning output as the current turn starting state. GRAPHFLOW consisted of three main layers; Encoding, Reasoning, and Prediction.

The Community Answer Generation method based on the Knowledge Graph (CAGKG) [133] was proposed to generate natural language answers automatically. CAGKG injected the user background in the search process of relevant knowledge entities to increase its efficiency. Also, to enhance the semantic understanding of the questions and answers, it utilized phrases and their relations that were extracted using a parsing tree. Then, the user's knowledge background and these semantic phrases were used to construct the Knowledge Graph (KG) to extract the relevant entities and convert them into natural language answers. Moreover, a new Phrase-based Answers Semantic Similarity Evaluation indicator (PASSE) was designed. PASSE exploited phrase overlay to concentrate on the text semantic similarity instead of exact literal matching of the generated answers.

The Coarse-grain Decomposition Fine-grain interaction (CGDe-FGIn) model was proposed in [107] that tackled the multi-hop reasoning by leveraging two strategies: Coarse-Grain

Decomposition (CGDe) strategy and Fine-Grained Interaction (FGIn) strategy. The former strategy decomposed complex questions into multiple simpler single-hop ones with no additional annotations. On the other hand, the latter strategy enhanced the fine-grained features, i.e., the word representations, to better comprehend and extract accurate sentences needed to answer the question.

The Query-Focused Extractor (QFE) model was developed in [108] for explainable multi-hop reasoning based on extractive summarization models. QFE utilized sequential multi-task learning for answer selection and evidence extraction. Its query-aware recurrent-based attention-guided structure benefited from the summarization models' evidence dependency and question coverage. QFE adaptively determined the required number of evidence sentences and thus, adaptively terminates the reasoning process by monitoring and evaluating the extracted evidence exact matchings and precision scores at each hop and reformulating the question accordingly.

Dynamically Fused Graph Network (DFGN), a graph-based multi-hop reasoning QAS, was proposed in [109] to mimic the human's step-by-step reasoning behavior. Five components were developed in DFGN. The first component was the relevant paragraph selection subnetwork, followed by the entity graph construction module with the entities as nodes and the entities' co-occurrences as edges. An encoding layer was the third component that operated on concatenating the questions and contexts that provided the token-level contexts. The fourth was a dynamic fusion block for multi-hop reasoning to gradually find the reasoning chain that contained the supporting entities for a given question by performing a dynamic graph attention mechanism on the entity embeddings iterated for multiple steps. Lastly, the prediction layer output the obtained supporting sentences, the start and end positions of the answer, and the answer type.

The Bidirectional Attention entity Graph convolutional network (BAG) was proposed in [110] to address the task of multi-hop reasoning. BAG leveraged the advantages of bidirectional attention operated in multi-level graph features. BAG began by constructing the entity graph by transforming the entities in the documents into connected nodes using two types of edges: cross-document edges and within-document edges. At this level, four types of multi-level features were aggregated to enrich the obtained representations: the isolated entities' token-level embeddings, the contextual-level features around these entities, NER, and POS manual features to capture the token's semantic properties. After that, the nodes' relation-aware representations, required to realize the multi-hop reasoning, were learned by importing the graph into graph convolutional networks. Finally, novel bidirectional attention between the generated multi-level features of the graph and a query was performed to learn a query-aware representation that was used to derive the final predictions of the answer.

A hierarchical Graph Network (HGN) for multi-hop QA was proposed in [111]. HGN is a multi-level fine-grained hierarchical graph framework. To capture the scattered clues from different paragraphs, heterogeneous nodes were constructed in a unified graph with different levels of granularity: questions, entities, sentences, and paragraphs, which were connected using seven types of edges. This helped HGN perform subsequent sub-tasks, including paragraph selection, supporting facts extraction, and answer prediction tasks, by capturing different semantic and structural information to provide the required supervision. HGN consisted of four modules. It began with the Graph Construction Module that operated on the identified relevant paragraphs. Then, it was followed by the Context Encoding Module to provide the initial contextualized representations of the graph nodes that were updated and propagated jointly using the graph attention-based message passing algorithm in the third Graph Reasoning Module to achieve the multi-hop reasoning. Finally, the Multi-task Prediction Module, where the different subtasks were conducted simultaneously.

Coarse and Fine Granularity Graph Network (CFGGN) was proposed in [112] for interpretable multi-hop reasoning. CFGGN fused and aggregated sentence and entity-level reasonings to learn more contextual representations. The coarse-grain module captured the sentence representations and filtered out the noisy and unrelated sentences, to the question, from the sentence graph to focus the search. In contrast, the fine-grain module constructed a dynamic entity graph. Moreover, the sentence-level analysis allowed the model to benefit from the non-entities that existed in the context to understand the question better. The two attention graphs were constructed separately, with their nodes masked using a confidence score that was step-wise updated. Two attention mechanisms were adopted to extract the semantic features: the bidirectional attention to capture query-to-context and context-to-query embeddings and highlight the relevancies, and the self-attention to capture the query's syntactic information. Finally, the prediction module outputted the supporting facts, the type, and the start and end position of the predicted answer.

Select, Answer and Explain (SAE) system was proposed for interpretable multi-hop reasoning in [113]. To decrease the distraction in the document search space, the unrelated document filtering process was carried out using a classifier coupled with a novel pairwise learning-to-rank loss. After that, the remaining selected documents were fed simultaneously into the “answer and explain” graph-based model to generate two outputs: the answer and the corresponding supporting facts. This was generated using a multi-task learning objective with mixed attention-based interactions that operated on two levels; the token level to predict the answer and the sentence level to select the supporting facts. The nodes in the attentional graph were the contextual sentence embeddings with three types of edges; within-document edges to capture the document's global information and two types of cross-document edges that connect the nodes with shared named entities and noun phrases together and with the question.

An adaptation of the transformer inspired by Graph Neural Network (GNN) was proposed in [114] with eXtra Hop attention, called Transformer-XH. It combined the former advantage of text understanding and the latter of modeling structure to guarantee the Transformer-XH with the data-driven ability to model structured texts in a unified model. Transformer-XH utilized two attention mechanisms: in-sequence attention, which was already adopted by the vanilla transformer, and eXtra Hop attention, which performed multi-evidence reasoning. It conducted the reasoning by jointly aggregating information across multiple documents according to their original natural structure rather than sequentially and constructing global contextualized representations.

GREASELM, an abbreviation of Graph REASONing Enhanced LM, is a model proposed in [115] that exchanged, interactively fused, and jointly reasoned over the encoded representations obtained from pre-trained LMs and KGs. The operation of GREASELM over multiple modality interactions allowed it to ground and link the contextual representations by the structured explicit world knowledge. GREASELM consisted of a unimodal transformer-based LM block, a GNN layer, and a cross-modal fusion component.

A graph-based multi-hop QAS called TransferNet was proposed in [116]. It aimed to tackle the performance optimization and answer interpretability simultaneously by supporting the interaction of two relation types in a unified framework: structured KG labels and free text in the textual relation graph. The former is more expensive to obtain since they are based on constrained manually defined predicates, while the latter is retrieved from a text corpus based on entity pair co-occurrences. The entities formed the nodes, while the associated relations were considered the edges. TransferNet inferred the answer through multi-hop reasoning by starting with the entity representing the question topic, then analyzing different question parts at each time step and deciding the most proper relation according to a conditioned entity

score of the query so far. After that, this computed score was transferred across the already activated relations in a differentiable fashion. This was repeated for different steps (hops) until reaching the target entity.

A unified two-phase framework that supports any-hop reasoning called IDRQA was introduced in [117] with Iterative Document Reranking (IDR) phase and a question answering phase. IDRQA used the TF-IDF method to retrieve the top possible related documents to answer a certain question. IDR formed the document attention graph by extracting these documents and question entities using the NER method and connecting multiple documents if they shared an entity. Then, a graph-based re-ranking model scored each supporting document according to the constructed graph to filter irrelevant documents and reduce noise. The re-ranking model also updated the question based on the extracted clue span from the retrieved documents to either form an extended version of the question and consequently retrieve more new documents and concatenate them to the graph or terminate the retrieval process. After that, the documents with the highest score were propagated along with the question to the reader model for answer extraction if it exists. This adaptive iterative process allowed IDRQA to handle questions with varying complexity instead of statically fixing the number of needed reasoning hops.

A Latent Execution Guided reasoning framework, LEGO, was introduced in [118]. LEGO adopted an iterative two-phase approach. The Query Synthesizer phase synthesized, in a context-aware step-wise manner, a reasoning action by performing a bottom-up guided search based on the so-far-obtained partial execution to efficiently grow the query tree step-by-step in the vast KG space. On the other hand, the Latent Space Executor phase executed the constructed reasoning action tree to address the missing information and update the embeddings.

To enhance the context retrieval process required in the conventional multi-hop reasoning task, especially for the open-domain questions, Multi-hop Dense Retriever (MDR) was proposed in [119]. The advantages of MDR were drawn from its ability to shrunken the exponentially growing search space with each reasoning step. This recursive framework employed dense retrieval methods such as Maximum Inner-Product Search (MIPS).

Table 13 lists the specifications of QASs dependent on graph attention neural models; most models tackled multi-hop reasoning or conversational answer prediction. As noted from the table, the recent pre-trained contextualized encoding is the dominant method for representing the words due to the task's difficulty.

3.6 Generative-based models

Cross-sentence context Aware Bidirectional LSTM model (CABIN) for community QA was proposed in [120]. It created context-aware representations by processing the question and answer together. Moreover, CABIN proposed two parallel attention mechanisms to operate on the sentence level to generate co-attention weights and on the interaction level to account for the adjacent words' relationships and similarities in the question and the answer. This attention mechanism is called the context information jump. These resulted attention-driven interactive sentence-aware representations automatically selected the salient positional representations for better relevancy matching between the question and answer using an adaptive similarity matrix.

The Knowledge-Enriched Answer Generator (KEAG) was proposed in [121] that incorporated external knowledge into the machine reading task. KEAG adaptively determined

when to utilize symbolic knowledge. It generated answers by flexibly exploiting and aggregating relevant facts from four different information sources: question, passage, vocabulary, and knowledge, using the source selector module. The advantage of including commonsense knowledge in KEAG was the ability to utilize the relevant information that was not explicitly mentioned in the context and generate answer words that were not mentioned in the vocabulary. KEAG was an extended sequence-to-sequence model with attention in which it had separate Bi-LSTM-based encoders for the passage and the question. Then, the single unidirectional LSTM decoder picked an answer word from the selector module according to the attention distribution and semantic relevancy at each time step.

Multi-Hop Pointer-Generator Model (MHPGM) was proposed in [122] to tackle the generative multi-hop question answering task based on commonsense knowledge by reading context, reasoning over disjoint information, and synthesizing proper coherent answers. MHPGM, a graph-based model, integrated multiple hops of bidirectional BiDAF attention [88] to iteratively update the context representation based on: query information, residual-based self-attention to resolve contexts' long-term dependencies and co-references, and a pointer-generator decoder that operated on the context to attend its entities for answer synthetization. To address the issue of implicit relations understanding and long-distance reasoning, external, background commonsense knowledge is needed. Thus, Necessary and Optional Information Cell (NOIC) was also proposed to fill the reasoning gaps between context hops via selecting grounded multi-hop commonsense knowledge paths from ConceptNet [141] graph via a selectively gated attention mechanism that used pointwise mutual information and term-frequency based scoring function.

A two-step approach for tackling open domain QA was proposed in [123]. At first, the potential supporting passages were retrieved using either sparse or dense representations. Then, each question with its selected passages and their titles were fed to a sequence-to-sequence generative model; each was coupled with a separate encoder independently to accelerate the processing and avoid limiting the number of allowed passages. Then, the concatenation of the retrieved passages representations was fed to the Fusion-in-Decoder to generate the answer by jointly aggregating and combining evidence collected from multiple passages according to their attention.

PATHFID is an extension of the generative Fusion-in-decoder and was proposed in [124] for multi-hop reasoning. PATHFID combined the reasoning paths, generated answers, and supporting facts jointly and explicitly. PATHFID was a sequence-to-sequence architecture that adopted a single sequence prediction task that eased the complex reasoning chain and eventually generated the answer in a structured way. This task aimed to produce a unified linearized representation of the hierarchical reasoning path of the retrieved supporting documents and facts by encoding the cross-passage interactions.

Table 14 lists the specifications of QASs that depend on generative models. The metrics to evaluate the reviewed works are usually more complex than those used for the extractive QASs.

3.7 Reinforcement learning-based models

A model that followed a hierarchical structure was proposed in [16], combining a coarse fast model and a computationally expensive recurrent-based model. They were aided with reinforcement learning to efficiently scale the QAS to deal with long documents. A few relevant sentences were retrieved from the document by the fast model to shorten and narrow the focus on limited sections of the document. Then, an in-depth investigation was done by

the RNN model to generate the final answer from those retrieved sentences jointly with the help of reinforcement learning by treating these sentences as latent variables.

The Reasoning Network (ReasoNet) was proposed in [125] based on multiple iterations over the input. ReasoNet applied different attention to each iteration; each iteration focused on a different input portion to obtain the required answer. The iteration's termination criteria were determined using reinforcement learning to determine whether to accept the reached results as an answer or to start another iteration to enhance it. Therefore, the number of iterations was dynamic and learned automatically based on the degree of difficulty of the question.

The Active Question Answering (AQA) model [126] consisted of three ingredients: a question reformulator, a black box QAS, and a candidate answer aggregator. It did not focus on the QAS itself but on enhancing its inputs and outputs using the reformulator and aggregator. These two components were trainable agents with the goal of finding a non-trivial interpretable reformulation of the question and outputting the best answer.

The Dynamic Fusion Network (DFN) for machine reading comprehension was proposed in [127]. The answer generation in DFN underwent two stages: dynamic multi-strategy attention and dynamic multi-step reasoning. Reinforcement learning was used to determine the best attention strategy and the optimal number of reasoning steps to adapt uniquely to each question type.

The end-to-end Reinforced Mnemonic Reader (RMR) [128] introduced a novel re-attention mechanism that refined the attention outputs in a multi-round alignment (between question and context) architecture by temporally memorizing past attentions to filter out the redundancy and deficiencies in these attention outputs and focus on the relevant portions. RMR also employed dynamic-critical reinforcement learning to enhance its supervised learning. Based on two sampling strategies: random and greedy inferences, a dynamic reward, and the baseline were decided to tackle the convergence suppression limitation in the traditional reinforcement learning algorithms by predicting a more acceptable answer. The architecture consisted of three main components: an encoder, an iterative reattention aligner, and finally, an answer pointer. The encoder built the contextual representation for both the question and the context jointly, then a multi-round alignment between the question and the context was done through three reattention mechanisms; an interactive aligner to attend the question with the context; a self-alignment to attend the context against itself; and an evidence collector to model the resulted representations with a Bi-LSTM. Finally, the answer span was predicted using the answer pointer network [139].

Table 15 lists the specifications of QASs that integrate the reinforcement learning concepts in their models.

3.8 Other models

The Relational Networks (RNs) [129] focused on relational reasoning. A feed-forward network was used to aggregate the correlations among the different entities in the data (the document, the question, and the answer) by implementing and applying similar functions to all of them. The RN considered all the permutations to validate that all the entities' relationships have been explored in the data.

A cascade approach for answer extraction was proposed in [130]. The model was the cascading of three lightweight submodels; each consisted of feed-forward networks with an attention mechanism, and each operated at a different level. In the first level, bag-of-embeddings representations of the question versus a candidate answer span and a candidate

answer span versus the words of the context surrounding it in the document were generated independently. Then, the second level submodel aligned the question words with the context words that contained the candidate's answer based on the first level's representations and used attention to select the best answer candidate. Finally, the third submodel aggregated the information or evidence from multiple mentions of answer candidate spans throughout the document to build a single answer representation.

The HyperQA [131] had a self-organizing, parameter-efficient neural network that had automatic latent hierarchies' discovery abilities. It replaced the Euclidean space with the Hyperbolic space to model the relationship between the embedding of the question and a candidate answer to form a pairwise ranking.

BLOCK AttentionN for Context prediction (BLANC) was proposed in [132] to address the discrepancy issue that occurred when the context contains multiple answer-text occurrences. BLANC was built based on two novel ideas: context word prediction and the block attention method. The former was done using the soft labeling method by calculating the word context-aware probabilities to the answer spans. At the same time, the latter identified the answer by reflecting the words' spatial localities and predicting the soft labels. Thus, BLANC predicted the answer-span correctly according to the context probability given the question.

The Knowledge Embedding-based Question Answering (KEQA) framework [105] proposed to answer a simple question asked in a natural language. The model operated on the KG embedding space to benefit from its preserved structures and relations; it jointly learned the question's head, predicate, and tail entities representations using an attention-based Bi-LSTM. Then, the shortest distance fact in the KG was retrieved as the answer based on a designed joint distance metric.

The Deep Fused Model (DFM) was proposed in [134], where a subject detection task and multi-level predicate matching task were trained jointly to learn the questions' semantic representation. The former focused on recognizing the question part that was more relevant to the predicate, while the latter was used to learn the shallow and deep semantic questions and predicate features through a gating mechanism.

Table 16 lists the specifications of QASs that could not be classified under any of the previous categories.

4 Famous QA datasets

One of the significant contributors to the success and good performance of a QAS is the knowledge accompanied with it, i.e., the existence of large, high-quality datasets to train and evaluate the system model. Some famous and widely used datasets in the QA domain are described next, in addition to the dataset's statistics and the performance of some of the reviewed papers that operated on these datasets to relatively compare their effectiveness. Since it is impossible to re-implement all the reviewed models in these papers, we only report the results published in their original papers. The reviewed datasets were classified according to their answer type, as shown in Fig. 4.

4.1 Factoid datasets

- **TERC QA tracks 8 to 13:**³ A domain-independent question-answer re-ranking dataset proposed in [15]. The dataset contains a set of factoid questions with a list of corresponding

³ Download link: <https://trec.nist.gov/data/qa.html>, Date of Access: 5th Jun, 2022.

potential answers limited to a single sentence. These factoid questions are collected from the 8–13 Text Retrieval Conference (TREC)⁴ data tracks and classified into ‘who,’ ‘what,’ ‘where,’ ‘when,’ and ‘why’ questions. The TREC dataset consists of questions, answer patterns, and a document pool. A combination of overlapping non-stop word counts and pattern matching is used for answering candidates’ ranking according to how relative they are to the question. TREC 8–12 tracks questions are used for training, while TREC 13 is used for development and testing. The correctness of the candidate answer selection of all the TREC 13 questions and the first 100 questions from TREC 8–12 is provided by manual judgment. Therefore, there are two training data sets: TRAIN and TRAIN-ALL. TRAIN consists of QA pairs with candidate answers that have been manually judged and annotated. At the same time, TRAINALL contains the pairs that are automatically judged by matching the candidate answers against regular expressions of the answer patterns provided by TREC. The TRAINALL dataset is larger and noisier. Table 17 summarizes the TREC dataset statistics. According to [48], two versions of the TREC dataset are clean and raw, giving different and non-comparable results. Both TREC versions share the same training set; however, the questions from the development and test sets that have no candidate answers or have no negative candidate sentences are removed from the clean TREC. Table 18 shows the results of the performance of the reviewed models in this survey that operate on the raw version of TREC. As noted from the reported results of these models, the new variant of BERT, RoBERTa, and the TL principle achieved the best performance.

- **WikiQA:**⁵ This is a domain-independent factoid QA dataset collected from crowd-sourced annotations of the English Wikipedia passages and Bing search query logs [142]. It is a sentence-level-based dataset with question-like queries where the required task is to decide if each candidate sentence provides the answer to the query by ranking their correctness likelihood. The frequent queries, issued by five or more distinct users extracted from Bing query logs and led to the retrieval of Wikipedia pages, were selected as questions. On the other hand, the sentences of the corresponding summary section of these pages were used as the candidate answers. Three crowdworkers then verified the correctness of these answers to identify the correct one. WikiQA dataset may contain questions with more than one correct answer or with no correct answer at all. This encourages research in answer triggering. The relevant statistics of the WikiQA dataset and the top-performing models among the reviewed papers in this survey are shown in Tables 19 and 20, respectively. As can be noted from the results in Table 20, the MAP and MRR metrics are adopted for WikiQA evaluation.
- **InsuranceQA:**⁶ It is a domain-specific dataset containing real-world question and answer pairs in the insurance domain [47] gathered from the internet. This dataset is divided into four parts: train, development, test 1, and test 2. It contains 17,487 questions and 24,981 unique answers; since the questions in this dataset can have multiple correct answers, this formed a candidate answer pool from which a random answer is selected. A more detailed description is given in Table 21.

⁴ The Text REtrieval Conference (TREC) is a series of workshops that provides the needed infrastructure for text retrieval methodologies with large-scale evaluation.

⁵ Download link: <https://download.microsoft.com/download/E/5/F/E5FCFCEE-7005-4814-853D-DAA7C66507E0/WikiQACorpus.zip>, Date of Access: 5th Jun, 2022.

⁶ Download link: <https://github.com/shuzi/insuranceQA>, Date of Access: 5th Jun, 2022.

Table 17 Statistics of the raw TREC dataset

	Train-all	Train	Dev	Test
# Questions	1229	94	82	100
# QA pairs	53,417	4718	1148	1517
Judgments types	Automatic	Manual	Manual	Manual
%correct (PosRate)	12%	7.4%	19.3%	18.7%

Table 18 The results of some of the reviewed papers on the TREC dataset

QA model	Publishing year	MAP	MRR
[60] RoBERTa+ TANDA	2019	0.9430	0.9740
[120] CABIN	2020	0.8375	0.8845
[69] Dynamic-Clip Attention	2017	0.8210	0.8990
[51] BiMPM	2017	0.8020	0.8750
[48] Rank MP-CNN	2016	0.7800	0.8300
[131] HyperQA	2018	0.7700	0.8250
[52] HD-LSTM	2017	0.7500	0.8153
[74] aNMM	2016	0.7495	0.8109
[71] CNN+ attention+ matching matrix	2018	0.7113	0.7803

Table 19 Statistics of the WikiQA dataset

	Train	Dev	Test	Total
# Questions	2118	296	633	3047
# Candidate sentences	20,360	2733	6165	29,258
# Correct answers	1040	140	293	1473
Average length of questions	7.16	7.23	7.26	7.18
Average length of sentences	25.29	24.59	24.95	25.15
# Questions w/o correct answers	1245	170	390	1805

4.2 Non-factoid datasets

- **WikiPassageQA:**⁷ It contains the top 863 Wikipedia pages from the Open Wikipedia Ranking created by Amazon Mechanical Turk [67]. Multiple queries are associated with each Wikipedia page; their answers have varying lengths within the Wikipedia document, with a total of 4165 queries. Table 22 presents the statistics of the dataset.

⁷ Download link: <https://ciir.cs.umass.edu/downloads/wikipassageqa/WikiPassageQA.zip>, Date of Access: 5th Jun, 2022.

Table 20 The results of some of the reviewed papers on the WikiQA dataset

QA model	Publishing year	MAP	MRR
[60] RoBERTa +TANDA	2019	0.9200	0.9330
[50] Question Classification	2018	0.8625	0.8362
[85] MCAN	2018	0.8380	0.9040
[70] Comp-Clip + LM + LC +TL	2019	0.8340	0.8480
[86] TL + BiDAF	2017	0.8320	0.8458
[27] BERT (single model)	2018	0.8130	0.8280
[120] CABIN	2020	0.7520	0.7653
[69] Dynamic-Clip Attention	2017	0.7540	0.7640
[51] BiMPM	2017	0.7180	0.7310
[131] HyperQA	2018	0.7120	0.7270
[73] KV-MemNN	2016	0.7069	0.7265
[49] L.D.C.	2017	0.7058	0.7226
[48] Rank MP-CNN	2016	0.7010	0.7180
[75] QA-TF _{AP}	2019	0.6941	0.7077

Table 21 Statistics about the InsuranceQA dataset. A total of 2,386,749 words in the answer text, which corresponds to 24,981 unique answers

	#Questions	#Answers	Question word count
Train	12,887	18,540	92,095
Dev	1000	1454	7158
Test1	1800	2616	12,893
Test2	1800	2593	12,905

- **SemEval CQA (Task 3A):** It is a real-world dataset obtained from Qatar Living Forums and is provided by [143]. There is a thread of associated comments for each community question, with each comment representing a possible answer to the question. These comments are subsequently ranked “Good,” “Potentially Useful,” and “Bad” according to the degree of relevance to the question. The corpus in SemEval-2015 consists of 3229 questions: 2600 for training, 300 for development, and 329 for testing, with 20,162 total comments in the dataset. On the other hand, the SemEval-2016 corpus contains 800, 200, and 300 questions for training, development, and testing, respectively, with ten comments associated with each question.

4.3 Reading comprehension datasets

- **SQuAD:**⁸ It stands for “Stanford Question Answering Dataset,” and it is introduced in [144]. It is a large-scale reading comprehension dataset consisting of a mixture of factoid and non-factoid questions. The answers range from a single word to long, variable-length phrases. Thus, SQuAD requires inferring the required answer through different forms

⁸ Download link: <https://deepai.org/dataset/squad>, Date of Access: 5th Jun, 2022.

Table 22 Statistics of the WikiPassageQA dataset

	Train	Dev	Test	Total
# Questions	3332	417	416	4165
# Candidate passages	194,314	25,841	23,981	244,136
Average length of questions	9.52	9.69	9.44	9.53
Average length of answer passages	133.092	134.132	132.650	133.158

of logical reasoning. The main advantage of this dataset is being realistic since humans manually crowdsource it. It comprises 536 English Wikipedia articles with more than 100 K related question–answer pairs. Each crowdworker was asked to answer up to five questions about a Wikipedia passage by highlighting the answer in the passage. Table 23 shows the statistics for the SQuAD dataset. The test data are hidden and evaluated by the dataset organizers only. Each question has several ground-truth answers provided by different people. The original version of SQuAD is called SQuAD v1.1; the results of the performance of the reviewed models in this survey on this dataset are presented in Table 24. A newer version of the dataset, called SQuAD v2.0, combines the previous version and an additional 50 K unanswerable questions written by crowd workers to test the system’s ability to determine if the associated paragraph cannot provide an answer.

- **CNN/Daily Mail:**⁹ The two datasets are Cloze-style reading comprehension datasets proposed by [91]. The training set contains online news articles from the CNN and Daily Mail websites; 92,579 CNN articles were collected from 2007 to April 2015 with 387,420 associated questions, and 219,506 Daily Mail articles were collected from 2010 to April 2015 with 997,467 questions. On the other hand, the validation and testing sets were collected from March 2015 and April 2015, respectively. Each of these articles is associated with short abstractive summary statements describing the articles. The questions are generated synthetically by replacing a named entity in these summaries with a placeholder token. This removed word is considered the answer to that question. The article’s body forms the document in the resulting document–query–answer triples that form the corpus. To force the QAS to rely solely on the context when generating the answer, all the named entities in the documents were replaced by anonymous tokens. Tables 25 and 26 show the statistics and the performance of some of the reviewed models on the CNN and Daily Mail datasets, respectively.
- **bAbI:**¹⁰ The bAbI dataset is released by Facebook [145]. The main advantage of this dataset is its limited vocabulary since it is composed of synthetically generated stories about activities in a simulated world. Another appealing feature of this dataset is the wide range of available reasoning aspects that aims to test a specific capability in a QAS; this is achieved through the set of 20 QA tasks in this dataset. Each of these tasks corresponds to a different type of question; these tasks range from single supporting fact questions, two supporting fact questions, yes–no questions, counting questions, and positional reasoning. The tasks require a triplet input consisting of a variable-length passage of text context, a task-dependent question, and an answer.

⁹ Download link: <https://github.com/abisee/cnn-dailymail>, Date of Access: 5th Jun, 2022.

¹⁰ Download link: <https://research.fb.com/downloads/babi/>, Date of Access: 5th Jun, 2022.

- **RACE:**¹¹ It stands for “**Re**Ading **Com**prehension **Da**taset **F**rom **E**xaminations.” It is a large-scale multiple-choice reading comprehension dataset [146]. It is divided into two categories that vary in difficulty level; RACE-M and RACE-H, collected from Chinese students’ middle and high school English exams, respectively. Each question in the dataset contains four possible options with only one correct option. The combined RACE dataset contains 27,933 passages and 97,687 questions for the training, development, and testing. According to [146], the questions in RACE are targeted to test human comprehension skills and generated by domain experts; therefore, they require more reasoning capabilities, such as summarizing, inference, and deduction, than those of SQuAD. There are five question classes: Word matching, Paraphrasing, Single-sentence reasoning, Multi-sentence reasoning, and Insufficient/Ambiguous. Tables 27 and 28 show the statistics and the top-performing models of the RACE dataset.
- **ReCoRD:**¹² It stands for “**Re**ading **Com**prehension with **Com**monsense **Re**asoning **Da**taset,” and it is introduced in [147]. It is a large-scale dataset that consists of passage-question-answer tuples that are constructed from the CNN and Daily Mail news articles as follows: the passage is taken from the first few paragraphs of a news article that has marked named entities; the question that has a missing entity is taken from the rest of the article excluding the part taken to form the passage, and the answer is one of the marked entities in the passage. The process of finding the answer requires both external knowledge and commonsense reasoning. It contains more than 70,000 news articles and more than 120,000 questions, and corresponding answer rs.
- **TriviaQA:**¹³ It is a large-scale dataset for the reading comprehension task proposed in [148]. It is composed of a collection of 96 k trivia question-answer pairs collected from 14 trivia and quiz-league websites. The documents used to provide evidence of the answers to the questions, i.e., contains the answer, are collected independently from the web search or Wikipedia, and each document is pruned to 1200 words. However, since the document collection is separated from the question-answer pair generation, these documents may not contain the information needed to infer the answer correctly, but it allows to control the bias in question style and content. The TriviaQA documents are considered longer than the ones used in SQuAD Dataset. More statistical details of the dataset are introduced in Table 29.
- **NarrativeQA:**¹⁴ It is a large-scale free-form reading comprehension dataset developed in [149]. The context of the crowdsourced human questions came from the abstractive summaries of stories collected from two sources: books fetched from Project Gutenberg and movie scripts scraped from the web. These summaries are obtained from Wikipedia using articles’ titles and verified using human annotators. This resulted in a total of 47 K question-answer pairs and about 1.6 K stories’ summaries with around 30 questions each. The questions were generated based on the contents of the summaries. However, there are two tasks based on the place to explore to generate the answer: NarrativeQA-Summary and NarrativeQA-Story, with the latter being more challenging. The free-form generated answers must be grammatically correct and complete and can vary in length from a single word to multiple sentences. Table 30 gives detailed statistics about the dataset.

¹¹ Download link: <https://www.cs.cmu.edu/~glai1/data/race/>, Date of Access: 5th Jun, 2022.

¹² Download link: <https://sheng-z.github.io/ReCoRD-explorer/>, Date of Access: 5th Jun, 2022.

¹³ Download link: <http://nlp.cs.washington.edu/triviaqa/>, Date of Access: 5th Jun, 2022.

¹⁴ Download link: <https://github.com/deepmind/narrativeqa>, Date of Access: 5th Jun, 2022.

- **MS-MARCO:**¹⁵ It is a large-scale free-form reading comprehension dataset proposed in [150]. The Bing-search user queries are considered the questions, while the real top retrieved web documents are considered the corresponding contextual passages to obtain the answers from. There is no guarantee that the question is answerable; it may have zero, one, or multiple human-generated answers. It contains 1 M questions sampled from the search queries paired with an average of ten relevant passages. The MARCO dataset offers the questions and their types, the answers and a well-formed version of it, and the context document used in the answer generation.
- **HOTPOTQA:**¹⁶ It is a large-scale span-based multi-hop QA dataset proposed in [151]. HOTPOTQA questions were collected by crowdsourcing based on Wikipedia articles using Amazon Mechanical Turk. Crowdworkers were given paragraphs of a pair of Wikipedia documents and asked to generate questions that could be answered using reasoning about both paragraphs to achieve the multi-step reasoning. The crowdworkers were also asked to provide, along with the answer, the supporting facts they used to reach that answer to be part of the dataset. The human-annotated sentence-level supporting facts provided by HotpotQA achieve multiple goals: answer reasoning explainability, intermediate QAS performance evaluation, and providing strong supervision to QAS. The questions in HotpotQA are diverse and require the exploration and reasoning over supporting facts that exist in multiple documents to answer a specific question. These questions exploited two different reasoning types: bridge and comparison. The questions that fall under the bridge category first identify the bride (linking) entities that lead to the answer. On the other hand, the comparison questions compare two entities that share a common category to test the QAS understanding of the language, entities' properties, and common concepts, since providing answers to questions usually requires numerical and arithmetic evaluation. Therefore, the answer candidates in HotpotQA are text spans or Yes/No. HOTPOTQA contains approximately 113 K question–answer pairs divided according to their difficulty levels to easy (single-hop), medium, and hard multi-hop questions, as shown in Table 31. Moreover, the HotpotQA dataset contains two benchmark tasks with different settings: Distractor and FullWiki. The two tasks share the same question sets, and each question context consists of ten paragraphs; however, they differ in the provided context. In the Distractor setting, two gold paragraphs with ground truth answers equipped with supporting facts are used to construct the question–answer pair, in addition to eight distractor paragraphs collected from Wikipedia that provided the potential best-matching paragraphs to answer that question. Since the distractor paragraphs introduced noise in the answer exploration process, this task tests the QAS abilities in denoising and reasoning. On the other hand, in the FullWiki setting, all ten paragraphs of each question are retrieved by examining the first paragraphs of all Wikipedia documents without providing any gold ones. Thus, this task is more challenging and tests the QAS's ability to locate, retrieve and reason over the relevant facts. Table 32 shows the performance of the reviewed papers in this survey that tested this dataset.

¹⁵ Download link: <https://github.com/deepmind/narrativeqa>, Date of Access: 5th Jun, 2022.

¹⁶ Download link: <https://hotpotqa.github.io/>, Date of Access: 5th Jun, 2022.

Table 23 Statistics of SQuAD v1.1 dataset

	Train set	Development set
# Articles	442	48
# Questions	87,599	10,570
# Passages	18,896	2067
# Answers	78,599	34,726
Questions length	11.4	11.5
Passages length	140.3	144.5
Answers length	3.5	3.3

Table 24 The results of some of the reviewed papers on the SQuAD v1.1 dataset

QA model	Publishing year	Test set	
		EM	F1
Human performance ^a	–	82.304	91.221
[33] ANNA Large	2022	90.6	95.7
[29] LUKE	2020	90.2	95.4
[31] XLNET large	2020	89.9	95.1
[132] BLANC _{large}	2020	87.3	93.4
[27] BERT large(ensemble model ^b + TriviaQA)	2018	87.4	93.2
[58] KT-NET (single model)	2019	85.9	92.4
[27] BERT large	2018	85.1	91.8
[72] QANet (ensemble model)	2018	82.7	89.0
[84] SLQA+ (ensemble model)	2018	82.4	88.6
[128] RMR (ensemble model)	2017	82.3	88.5
[28] BiDAF + Self-Attention + ELMo (ensemble model)	2018	81.0	87.4
[79] Memoreader + ELMo	2018	79.7	86.7
[89] SAN	2017	79.6	86.5
[82] DCN+ (ensemble model)	2017	78.9	86.0
[99] FusionNet (ensemble model)	2017	78.8	85.9
[104] GF-Net	2020	77.8	85.0
[78] MEMEN (ensemble model)	2017	76.9	84.0
[98] phaseCond	2017	76.1	84.0
[97] R-NET (ensemble model)	2017	75.9	82.9
[125] ReasoNet (ensemble model)	2017	73.4	81.8
[88] BiDAF(ensemble model)	2016	73.3	81.1
[80] DCN (ensemble model)	2016	71.6	80.4
[140] Ruminating Reader (single model)	2017	70.6	79.4
[94] Match-LSTM with Answer-Pointer (ensemble model)	2016	67.9	77.0

^aObtained from the leaderboard performance at the time of writing (5th Jun, 2022). <https://rajpurkar.github.io/SQuAD-explorer/>

^bEnsemble model means several identical models (except for the initial random parameters) are used in the testing, and the one with highest sum of confidence scores is reported

Table 25 Statistics about the CNN/ Daily Mail datasets

	CNN			Daily mail		
	Train	Valid	Test	Train	Valid	Test
# Months	95	1	1	56	1	1
# Documents	90,266	1220	1093	196,961	12,148	10,397
# Queries	380,298	3924	3198	879,450	64,835	53,182
Max # entities	527	187	396	371	232	245
Average # entities	26.4	26.5	24.5	26.5	25.5	26.0
Average # tokens	762	763	716	813	774	780
Vocabulary size	118,497	208,045				

Table 26 The results of some reviewed papers on CNN and Daily Mail datasets

QA model	Publishing year	CNN	Daily mail
[96] GA Reader	2017	77.9	80.9
[88] BiDAF	2018	76.9	79.6
[92] AS reader (average ensemble)	2016	75.4	77.1
[92] AS reader (greedy ensemble)	2016	74.8	77.7
[125] ReasoNet	2017	74.7	76.6
[95] AoA reader	2016	74.4	-
[93] Epireader	2016	74.0	-
[91] Impatient Reader	2015	63.8	68.0
[91] Attentive Reader	2015	63.0	69.0

Table 27 Statistics about the three datasets that form the RACE-M, RACE-H, and RACE

	RACE-M			RACE-H			RACE		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
# Passages	6409	368	362	18,728	1021	1045	25,137	1389	1407
# Questions	25,421	1436	1436	62,445	3451	3498	87,866	4887	4934
Passage length	231.1	353.1	321.9						
Question length	9.0	10.4	10.0						
Option length	3.9	5.8	5.3						
Vocabulary size	32,811	125,120	136,629						

Table 28 The results of some of the reviewed papers on the RACE dataset

Model	Publishing year	Accuracy
Human Performance: Amazon Mechanical Turker	–	73.3
Human Performance: Human Ceiling Performance	–	94.5
[59] Megatron-BERT (ensemble)	2019	90.9
[61] ALBERT + DUMA (ensemble)	2020	89.5
[36] ALBERT (ensemble)	2019	89.4
[40] DeBERTa _{LARGE}	2020	86.8
[36] ALBERT _{XXLARGE}	2019	86.5
[31] XLNET _{LARGE}	2020	85.4
[31] XLNET _{BASE}	2020	84.0
[39] RoBERTa _{LARGE}	2019	83.2
[36] ALBERT _{LARGE}	2019	75.2
[57] OCN LARGE (ensemble)	2019	73.5
[27] BERT _{LARGE}	2018	72.0
[40] DeBERTa _{BASE}	2020	71.7
[27] BERT _{BASE}	2018	65.0
[127] DFN (ensemble)	2017	51.2
[152] Hierarchical Co-Matching	2018	50.4
[100] Hierarchical Attention	2018	46.0

Table 29 Statistics about the TriviaQA dataset

# QA pairs	95,956
# Unique answers	40,478
# Evidence documents	662,659
Average question length (word)	14
Average document length (word)	2895

Table 30 Statistics about the NarrativeQA dataset

	Train	Valid	Test
# Documents	1102	115	355
# QA pair	32,747	3461	10,557
Average # token in summaries	659	638	654
Average # token in stories	62,528	62,743	57,780
Average # token in questions	9.83	9.69	9.85
Average # token in answers	4.73	4.60	4.72

Table 31 Statistics about the HotpotQA dataset

# QA pairs	Train		Development	Test/distractor	Test/FullWiki	Total
	Single-Hop	Multi-Hop (medium)				
Question Nature	18,089	56,814	15,661	7405	7405	112,779
		Multi-Hop (hard)	Multi-Hop (hard)	Multi-Hop (hard)	Multi-Hop (hard)	

Table 32 The results of some of the reviewed papers on the HotpotQA dataset

QA model	Publishing year	Answer		Supporting facts		Joint	
		EM	F1	EM	F1	EM	F1
<i>FullWiki Task/Test Set</i>							
[119] MDR + ELECTRA Reader	2021	62.3	75.3	57.5	80.9	41.8	41.8
[111] HGN + ALBERT _{xxlarge-v2}	2020	59.74	71.41	51.03	77.37	37.92	62.26
[117] IDRQA	2021	62.5	75.9	51.0	78.9	36.0	63.9
[114] Transformer-XH	2020	51.6	64.1	40.9	71.4	26.1	51.3
[108] QFE	2019	28.66	38.06	14.20	44.35	8.69	23.1
<i>Distractor Task/Test Set</i>							
[64] FE2H on ALBERT	2022	71.89	84.44	64.98	89.14	50.04	76.54
[63] S2G+EGA	2021	70.92	83.44	63.86	88.68	48.76	75.47
[64] FE2H on ELECTRA	2022	69.54	82.69	64.78	88.71	48.46	74.90
[111] HGN	2020	69.22	82.19	62.76	88.47	47.11	74.21
[32] BIGBIRD	2020	68.12	81.18	63.25	89.09	46.40	73.62
[113] SAE _{large}	2020	66.92	79.62	61.53	86.86	45.36	71.45
[108] QFE	2020	53.86	68.06	57.75	84.49	34.63	59.61
[112] GFGGN	2020	56.29	70.11	52.94	82.46	34.61	60.70
[109] DFGN	2019	56.31	69.69	51.50	81.62	33.62	59.82
[107] CGDe-FGIn model	2021	50.89	65.41	39.47	79.83	23.08	54.51

4.4 Conversational datasets

- **CoQA:**¹⁷ It stands for “**C**onversational **Q**uestion **A**nswering.” It is a large-scale dataset proposed in [17]. It contains 8 k conversations from seven diverse domains with 127 K associated questions with answers: Wikipedia, Reddit, news and science articles, child stories, literature, and middle and high school English exams. Two crowd workers are required to chat in the form of questions and answers about a passage from a specific domain to construct a conversation, with the condition that the question words must differ from the passage words to obtain lexical diversity. In contrast, the free-form answer words were restricted to include passage words, limiting the candidate answer possibilities. The questions in this dataset require revising the conversation history and having some reasoning skills, such as co-reference and pragmatic reasoning, to generate a free-form answer by highlighting a span in the passage as the rationale of this answer.
- **QuAC:**¹⁸ It stands for “**Q**uestion **A**nswering in **C**ontext.” It is a large-scale conversational reading comprehension dataset proposed in [153]. It contains 13.5 k conversations and 98 k questions that take the form of student-teacher interactive dialog about Wikipedia articles. The answers are text spans selected from the context, unlike the answers in CoQA, with a maximum of fifteen tokens each. QuAC questions have different natures; they are contextually specific, open-ended, or sometimes unanswerable from the context.

¹⁷ Download link: <https://microsoft.github.io/msmarco/>, Date of Access: 5th Jun, 2022.

¹⁸ Download link: <https://quac.ai/>, Date of Access: 5th Jun, 2022.

5 Discussion

Figure 10 presents the most adopted deep learning technique each year, starting from 2018 till 2022, with the most experimented QA dataset in that year according to the reviewed works in this survey.

5.1 Toward end-to-end learning

As can be highlighted from the vast amount of research in the past 6 years devoted to QA and deep learning, deep learning proves its competitiveness and effectiveness in leading the search in the challenging tasks of NLP, such as Question Answering. A noticeable observation is that most of the reviewed QA models do not depend on linguistic resources but mainly rely on the deep models' strength in automatically analyzing and processing the data, as proven by the many proposed end-to-end systems. However, some challenges still face any model that is based on deep learning, such as the lack of the theoretical foundation of the deep learning models; the lack of interpretability of the model [7]; and the millions of parameters that must be tuned during the learning procedure to produce good quality results. Moreover, deep learning models are data-hungry; the more data are fed to them, the more vocabulary size the model must learn. As a sequence, the dataset may become insufficient to digest these vocabularies.

Furthermore, the QA field has lately witnessed the development of systems that rely on large-scale KBs, such as Freebase, to provide the structured knowledge resources needed, especially for factoid-based systems. For instance, incorporating commonsense knowledge, such as ConceptNet relations, improves the model performance to leverage the human ways of reasoning and answering real-world questions.

5.2 Toward the dominant architecture

Despite its impressive success and popularity in computer vision, there are few attempts to use CNN in the question answering field. This is mainly because CNN captures local spatial dependencies, not the semantic matching between the question and the answer, unlike the RNN, which captures sequential dependencies. This reason makes CNN inefficient in the semantic matching between the question–answer pair because of the complex linguistic properties of the natural languages. Moreover, longer text sequences require a larger network size, consequently increasing the computational cost. The CNN-based models are usually used as feature representation vector encoders, followed by a similarity metric to measure the semantic similarity between the extracted high-level features of the questions and answers. However, according to the trend followed by the investigated papers, CNN-based models seem to be outdated recently.

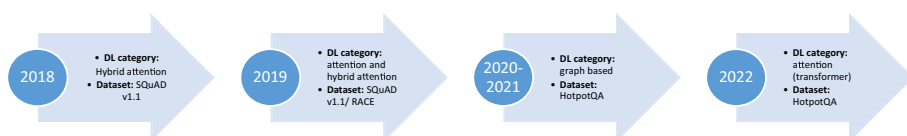


Fig. 10 The most widely used DL technique and dataset each year

Due to RNNs ability to tackle the sequential data by learning the temporal dependencies, they capture the order of the words in the text. However, RNNs suffer from some limitations; RNNs are problematic to train and require a long training time. Moreover, they suffer from the vanishing or exploding gradient problem since they use backpropagation. Another limitation is that RNNs can only capture short-term dependencies in sequential data and fail to capture long-term dependencies due to their limited memory, especially with the increase in the number of time steps; in other words, RNNs become computational bottlenecks when processing long sequences of text, which limits their performance.

However, as can be deduced from the previous tables that discussed the recent deep learning models in QASs, the popular models in this field are the famous variants of the recurrent networks: GRU and LSTM, and the memory-based network with an attention mechanism. The first two types allow the network to handle longer text and learn longer dependencies, while the third type allows the network to focus on the relevant facts from the input text. The LSTM was proposed to solve the limitations of the classical RNN model by keeping the gradient steep enough by preserving the errors propagated through time and layers to allow the recurrent network to learn over multiple time steps. Nevertheless, the main problem of the LSTM in the QA field is that there is no interaction between the question and the answer terms. Therefore, the LSTM model is more suitable for the simple QAS that does not demand high relational power. On the other hand, introducing memory to the network helps to reason over a simple fact that a single iteration can solve over the input context. The slightly more complex reasoning is solved by allowing multiple iterations over the context to refine the results using dynamic memory networks. However, the complex and memory-based deep learning models introduce high computational costs that may be unsuitable for practical application.

The attention mechanisms' powerfulness lies in their ability to jointly capture the relationship between the question and the context instead of learning them separately and fusing them later. Consequently, identifying the correlated words and avoiding discarding any important information in the text that may be needed for better extraction of the answer. Therefore, the attention mechanisms assist the combined deep learning model by focusing more on the specific and relevant parts of the context that are important for the question. Furthermore, the learned attention weights are gaining recent interest in the latest research as an attempt to offer explainability and interpretability to their QAS. Another observation from reviewing the research works is that the focus in recent years has shifted toward the models built solely on attention mechanisms, especially self-attention models like transformer-based ones. These models are also accompanied by dynamic LMs like BERT and its variants. The efficiency of the transformers lies in their parallelization ability to compute self-attention for the words in the text, which makes their hardware optimization more manageable. Figure 10 shows that QAS's dominant deep learning architecture is attention-based, either operated alone or hybrid with recurrency.

The generative models were implemented based on the sequence-to-sequence learning concept applied to the encoder–decoder framework. This type of QAS is challenging since many acceptable answers are linguistically and factually correct, with no single ground truth answer to make as a reference for evaluation. This puts more demand on developing evaluation metrics that can judge based on the semantic, not syntactic, similarities between the generated prediction and the ground truth answer.

Furthermore, as proven through analyzing the recently reviewed models, a successful and recent trend in enhancing the QASs combines deep learning methods with reinforcement learning for better optimization. Also, it can be clearly noted that the transfer learning concept is popular in the reviewed models since it helps the model to benefit from the knowledge

obtained from other architectures that are optimized and trained on large data to be used after a little modification on the dataset at hand. TL aims to overcome the expensive requirements of training a powerful deep learning model from scratch and eases the requirements of the need for large datasets.

Moreover, the previous tables that reported the best-performing models in some of the famous datasets showed that ensembling of multiple identical models trained in a single framework showed a superior performance but with the cost of introducing long inference times and extensive resources requirement. Some of the reviewed models even outperform the human performance on some datasets like SQuAD, as reported in Table 24.

5.3 Toward multi-hop reasoning

The early works in deep learning-based QASs begin by manipulating the simple extractive questions with factoid answers that needs only a single hop of reasoning. Then gradually, the task becomes more challenging and complex. The QASs are now assessed by their ability to generate answers in good natural language that require many ways of reasoning. Each station that the QASs endured is associated with datasets with certain characteristics to accomplish the task. The most investigated datasets in the latest years are SQuAD, RACE, and HotpotQA, as shown in Fig. 10. More recent attention and interest are given to datasets, such as TriviaQA or HotpotQA, that include questions that have complex interactions with the corresponding context that requires multi-step reasoning to determine and search for their answers. This is usually done using multi-hop or hierarchal attention to mimic the iterative human comprehension methods. Nonetheless, this may lead to attention redundancy and, consequently, deficiency.

Usually, two main research directions are followed by the multi-hop reasoning QAS. The first is to work bottom-up sequentially; in other words, based on the final answer, the model learns the answers' supporting facts that construct the intermediate reasoning chain. The second is to reduce the search space by filtering out the irrelevant context to the answer and reason about the remaining relevant passages using GNNs, usually integrated with attention. Both aspects have flaws and limitations; the first suffers from convergence and robustness issues since they handle all the context at once to find the answer, consequently increasing the search space rapidly and introducing noise that hinders the performance. On the other hand, the second usually sacrifices interpretability because the graph-based neural network handles the intermediate reasoning chain that suffers from the well-known deep learning BlackBox issue.

In general, three issues face the multi-hop reasoning research. The first is that the majority of the models define a fixed number of hops and reasoning steps in the answer prediction process. This was made based on assumptions obtained based on the used datasets. Therefore, the proposed QAS may face challenges when applied to other questions requiring more or less reasoning steps. A suggestion to address this issue is to use an adaptive mechanism that processes the question and dynamically determines the number of reasoning passes needed to answer it. The second is that many multi-hop QASs retrieve the most relevant passages to process them and extract the supporting facts. Retrieving many passages makes the search space large and noisy, making the answer extraction process hard. Contrarily retrieving a small number of relevant passages may result in missing some of the required supporting facts. This makes the number of retrieved passages a hyperparameter that must be tackled carefully. Finally, as the explainability and interpretability of QAS have become essential, especially with the existence of multi-hop reasoning datasets, an explanation of the intermediate steps

to better guide the search is a must to offer plausible justifications for the answer with more confidence and robustness. This is performed by predicting the supporting facts chain along with the final answer.

5.4 Toward better word representations

One of the most important key ingredients to successful and efficient QA models is the expressiveness degree of the word representations and how much this obtained representation reflects the correct semantic meaning of the word in a specific context. The word embedding used in the QASs is typically extracted from pre-trained Word2Vec or GloVe models trained previously using huge datasets. GloVe is more widely used than Word2Vec; the latter exploits the global information to define the target word embedding because of its co-occurrence matrix, unlike the former, that only utilizes the neighboring local information. However, the recent pre-trained contextualized language encoding, such as the BERT family, has become popular due to its good competitiveness in enhancing the combined architecture's performance. However, this is not always the case; the authors in [73] used Supervised Embedding trained on all of Wikipedia. On the other hand, some QASs depend on their dataset to initialize their word embedding, such as [86] and [95]. Other systems rely on the other neural network architecture to produce the word embedding, such as in [100], which used GRU to provide the word vectors. In summary, from the latest research, it seems that LMs will be the dominant and most used word representations, rather than the static word embeddings, in the future deep learning-based QA models.

Large LMs have recently been proposed, such as, among others, the autoregressive GPT-3 model [154] (Generative Pre-trained Transformer), which has 175 billion parameters, or the Megatron-Turing NLG model [155], which has 530 billion parameters. However, despite their boosts in performance, these models have massive computation and memory requirements; they use GPU Clusters consisting of hundreds of multi-GPU servers that cannot be affordable at individual levels. This may hamper their adoption and make their results hard to replicate. Therefore, we only focused on the relatively lighter LMs, and these large LMs are out of the scope of this survey.

On the contrary, many works adopted the concept of model compression to create more memory-efficient pre-trained LM architectures using either knowledge distillation techniques or efficient training mechanisms such as parameter sharing to cope with the existing infrastructure available for training.

5.5 Toward non-English question answering

As a final remark regarding the recent works, the vast majority of the proposed deep learning QASs are directed toward better-resourced languages such as the English language. The research in other languages is still shy and very basic. There are only a few deep learning-based QA proposed in non-English languages because it is a recent field of research, and the resources available for these languages are a few ill-structured. They lack well-balanced annotated datasets and a high-quality word embedding that can capture the semantics of the words. The datasets used in the few non-English proposed models are often based on translated English datasets, limiting the powerfulness of the deep learning models.

For example, an Arabic QAS based on RNN and aimed at factoid questions was presented in [156]. Another LSTM-based model for Community QA was proposed in [157] that operated on an Arabic-translated version of the SemEval 2016 Task 3-D from the medical

domain. An integration of CNN and RNN-based Arabic QAS [158] dealt with factoid questions that used an enriched version of the TALAA-AFAQ corpus with a manually translated subset of the UIUC question classification dataset. A Korean machine reading comprehension QAS that used a Simple Recurrent Unit (SRU)-based Self-Matching Network (S2-Net) was introduced in [159] and trained using a constructed Korean dataset collected from news and Wikipedia domains. A triple hybrid model that utilized CNN, attention mechanism, and RNN was proposed in [160] for the Chinese language Question answering, this model is called the Attention-Based BiGRU-CNN network (ABBC), and it was trained on two datasets: Wikipedia Chinese dataset and Fudan University Chinese question classification dataset. A Lattice-based CNN model (LCNs) to extract sentence-level features over word lattice in the Chinese language was proposed in [161] for short text matching; it also used two datasets: a document-based QA dataset called DBQA and a knowledge-based relation extraction dataset called KBRE.

6 Conclusion and future directions

This survey tried to enlist the famous recent research works in the QA field using deep learning models. It categorized and classified them based on multiple perspectives and discussed two possible taxonomies of different QASs. Also, several reviewed works were assessed, and their performance was ranked on well-known public datasets. This survey showed that the scope of the future research is broad, and the enhancement can be done at many levels; proposing richer and more powerful representations of the words, i.e., word embeddings and LMs; enhancing and improving the existing deep learning models by addressing the weakness and limitations of the previous models; proposing new models that are dedicated for the NLP tasks and can deal with raw and unsupervised data; proposing QASs that can deal with non-English languages; proposing new efficient training models; increasing the power of deep models with the reinforcement learning that integrate the environment interactions to enhance the predicted answer; and finally proposing a huge well-annotated, high-quality and structured dataset can play a critical role in providing a robust and accurate system for Question Answering. This survey attempted to be as comprehensive as possible to provide the understanding needed to shape the research in this field.

References

1. Toshevska M, Mirceva G, Jovanov M (2020) Question answering with deep learning: a survey. Faculty of Computer Science and Engineering Ss Cyril and Methodius University Skopje, Macedonia
2. Srba I, Bielikova M (2016) A comprehensive survey and classification of approaches for community question answering. *ACM Trans Web* 10:1–63. <https://doi.org/10.1145/2934687>
3. (2019) A survey on machine reading comprehension. *J Beijing Univ Posts Telecommun* 42:1
4. Huang Z, Xu S, Hu M et al (2020) Recent trends in deep learning based open-domain textual question answering systems. *IEEE Access* 8:94341–94356. <https://doi.org/10.1109/ACCESS.2020.2988903>
5. Palasundram K, Mohd Sharef N, Kasmiran KA, Azman A (2020) Enhancements to the sequence-to-sequence-based natural answer generation models. *IEEE Access* 8:45738–45752. <https://doi.org/10.1109/ACCESS.2020.2978551>
6. Abbasiantaeb Z, Momtazi S (2020) Text-based question answering from information retrieval and deep neural network perspectives: a survey. *Wiley Interdiscip Rev: Data Min Knowl Discov* 11:e1412. <https://doi.org/10.1002/widm.1412>
7. Li H (2018) Deep learning for natural language processing: advantages and challenges. *Natl Sci Rev* 5:24–26. <https://doi.org/10.1093/nsr/nwx110>

8. Xiang Y, Chen Q, Wang X, Qin Y (2017) Answer selection in community question answering via attentive neural networks. *IEEE Signal Process Lett* 24:505–509. <https://doi.org/10.1109/LSP.2017.2673123>
9. Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning in natural language processing. *IEEE Trans Neural Netw Learn Syst* 32:604–624
10. Vanitha G, Sanampudi SK, Guda V (2011) Approaches for question answering systems. *Int J Eng Sci Technol (IJEST)* 3:990–995
11. Riloff E, Wiebe J (2003) Learning extraction patterns for subjective expressions. In: *Proceedings of the 2003 conference on empirical methods in natural language processing (EMNLP)*, pp 105–112
12. Riloff E, Thelen M (2020) A rule-based question answering system for reading comprehension tests. In: *ANLP-NAACL 2000 workshop: reading comprehension tests as evaluation for computer-based language understanding systems*
13. Echihiabi A, Marcu D (2003) A noisy-channel approach to question answering. In: *Association for computational linguistics (ACL)*
14. Heie MH, Whittaker EWD, Furui S (2012) Question answering using statistical language modelling. *Comput Speech Lang* 26:193–209. <https://doi.org/10.1016/j.csl.2011.11.001>
15. Wang M, Smith NA, Mitamura T (2007) What is the jeopardy model? A quasi-synchronous grammar for QA. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp 22–32
16. Choi E, Hewlett D, Uszkoreit J et al (2017) Coarse-to-fine question answering for long documents. In: *Proceedings of the 55th annual meeting of the association for computational linguistics*, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 209–220
17. Reddy S, Chen D, Manning CD (2019) CoQA: a conversational question answering challenge. *Trans Assoc Comput Linguist* 7:249–266. https://doi.org/10.1162/tacl_a_00266
18. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18:1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
19. Severyn A, Moschitti A (2015) Learning to rank short text pairs with convolutional deep neural networks. In: *SIGIR 2015—proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. Association for Computing Machinery, Inc, pp 373–382
20. Miller GA (1995) WordNet. *Commun ACM* 38:39–41. <https://doi.org/10.1145/219717.219748>
21. Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: *COLING 2018, 27th international conference on computational linguistics*, pp 1638–1649
22. Adhikari A, Ram A, Tang R, Lin J (2019) DocBERT: BERT for document classification. *CoRR*. [arXiv:1904.08398](https://arxiv.org/abs/1904.08398)
23. Zhang H, Xu J, Wang J (2019) Pretraining-based natural language generation for text summarization. In: *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*. Association for Computational Linguistics, pp 789–797
24. Zhou C, Neubig G, Gu J (2019) Understanding knowledge distillation in non-autoregressive machine translation. In: *Proceedings of the 2019 international conference on learning representations*
25. Mikolov T, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems (NeurIPS)*, pp 3111–3119
26. Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp 1532–1543
27. Devlin J, Chang M-W, Lee K et al (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: *Annual conference of the North American chapter of the association for computational linguistics (NAACL)*
28. Peters M, Neumann M, Iyyer M et al (2018) Deep contextualized word representations. In: *Proceedings of the 2018 conference of the North American Chapter of the association for computational linguistics: human language technologies*, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 2227–2237
29. Yamada I, Asai A, Shindo H et al (2020) LUKE: deep contextualized entity representations with entity-aware self-attention. In: *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp 6442–6454
30. Clark K, Luong M-T, Le QV, Manning CD (2020) ELECTRA: Pre-training text encoders as discriminators rather than generators. In: *International conference on learning representations (ICLR)*
31. Yang Z, Dai Z, Yang Y et al (2019) XLNet: generalized autoregressive pretraining for language understanding. In: Wallach H, Larochelle H, Beygelzimer A et al (eds) *Advances in neural information processing systems*. Curran Associates Inc, Red Hook, NY

32. Zaheer M, Guruganesh G, Dubey KA et al (2020) Big bird: transformers for longer sequences. In: Larochelle H, Ranzato M, Hadsell R et al (eds) *Advances in neural information processing systems*. Curran Associates Inc, Red Hook, NY, pp 17283–17297
33. Jun C, Jang H, Sim M et al (2022) ANNA: enhanced language representation for question answering. In: *Proceedings of the 7th workshop on representation learning for NLP*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 121–132
34. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: *Advances in neural information processing systems*, pp 5998–6008
35. Goldberg Y (2019) Assessing BERT's syntactic abilities. CoRR. [arXiv:1901.05287](https://arxiv.org/abs/1901.05287)
36. Lan Z, Chen M, Goodman S et al (2019) ALBERT: a lite BERT for self-supervised learning of language representations. In: *International conference on learning representations (ICLR)*
37. Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR. [arXiv:1910.01108](https://arxiv.org/abs/1910.01108)
38. Wang W, Bi B, Yan M et al (2020) StructBERT: incorporating language structures into pre-training for deep language understanding. In: *8th international conference on learning representations (ICLR)*
39. Liu Y, Ott M, Goyal N et al (2019) RoBERTa: a robustly optimized BERT pretraining approach. CoRR. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692)
40. He P, Liu X, Gao J, Chen W (2021) DeBERTa: decoding-enhanced BERT with Disentangled Attention. In: *Proceedings of the 9th international conference on learning representations (ICLR)*
41. Jiang Z-H, Yu W, Zhou D et al (2020) ConvBERT: improving BERT with span-based dynamic convolution. In: Larochelle H, Ranzato M, Hadsell R et al (eds) *Advances in neural information processing systems*. Curran Associates Inc, Red Hook, NY, pp 12837–12848
42. Papineni K, Roukos S, Ward T, Zhu W-J (2002) BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting of the association for computational linguistics*
43. Banerjee S, Lavie A (2005) Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*
44. Lin C-Y (2004) ROUGE: a package for automatic evaluation of summaries. In: *Text summarization branches out: proceedings of the 2004 association for computational linguistics (ACL-04) Workshop*, Barcelona, Spain, pp 74–81
45. Zhang T, Kishore V, Wu F et al (2020) BERTScore: evaluating text generation with bert. In: *Proceedings of the international conference on learning representations (ICLR)*
46. Lee H, Yoon S, Dernoncourt F et al (2021) KPQA: a metric for generative question answering using keyphrase weights. In: *Proceedings of annual conference of the North American chapter of the association for computational linguistics (NAACL)*, pp 2105–2115
47. Feng M, Xiang B, Glass MR et al (2015) Applying deep learning to answer selection: a study and an open task. In: *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pp 813–820
48. Rao J, He H, Lin J (2016) Noise-contrastive estimation for answer selection with deep neural networks. In: *International conference on information and knowledge management, proceedings*. Association for Computing Machinery, pp 1913–1916
49. Wang Z, Mi H, Ittycheriah A (2016) Sentence similarity learning by lexical decomposition and composition. *COLING, Association for Computational Linguistics (ACL)*, pp 1340–1349
50. Madabushi HT, Lee M, Barnden J (2018) Integrating question classification and deep learning for improved answer selection. In: *Proceedings of the 27th international conference on computational linguistics*, pp 3283–3294
51. Wang Z, Hamza W, Florian R (2017) Bilateral multi-perspective matching for natural language sentences. In: *Proceedings of the twenty-sixth international joint conference on artificial intelligence*. International Joint Conferences on Artificial Intelligence Organization, California, pp 4144–4150
52. Tay Y, Phan MC, Tuan LA, Hui SC (2017) Learning to rank question answer pairs with holographic dual LSTM architecture. In: *SIGIR 2017—proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. Association for Computing Machinery, Inc, pp 695–704
53. Mihaylov T, Kozareva Z, Frank A (2017) Neural skill transfer from supervised language tasks to reading comprehension. *Workshop on learning with limited labeled data: weak supervision and beyond at NIPS*
54. di Gennaro G, Buonanno A, di Girolamo A et al (2020) Intent classification in question-answering using LSTM architectures. *Progr Artif Intell Neural Syst*. https://doi.org/10.1007/978-981-15-5093-5_11
55. Zhang L, Lin C, Zhou D et al (2021) A Bayesian end-to-end model with estimated uncertainties for simple question answering over knowledge bases. *Comput Speech Lang* 66:101167. <https://doi.org/10.1016/j.csl.2020.101167>

56. Hu M, Peng Y, Wei F et al (2018) Attention-guided answer distillation for machine reading comprehension. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 2077–2086
57. Ran Q, Li P, Hu W and Zhou J (2019) Option comparison network for multiple-choice reading comprehension. CoRR. arXiv:1903.03033
58. Yang A, Wang Q, Liu J et al (2019) Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. Association for Computational Linguistics, Stroudsburg, PA
59. Shoenybi M, Patwary M, Puri R et al (2019) Megatron-LM: training multi-billion parameter language models using model parallelism. CoRR. arXiv:1909.08053
60. Garg S, Vu T, Moschitti A (2020) TANDA: transfer and adapt pre-trained transformer models for answer sentence selection. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 7780–7788. <https://doi.org/10.1609/aaai.v34i05.6282>
61. Zhu P, Zhang Z, Zhao H, Li X (2022) DUMA: reading comprehension with transposition thinking. IEEE/ACM Trans Audio Speech Lang Process 30:269–279. <https://doi.org/10.1109/TASLP.2021.3138683>
62. Guu K, Lee K, Tung Z et al (2020) Retrieval augmented language model pre-training. In: III HD, Singh A (eds) Proceedings of the 37th international conference on machine learning. PMLR, pp 3929–3938
63. Wu B, Zhang Z, Zhao H (2021) Graph-free multi-hop reading comprehension: a select-to-guide strategy. CoRR. arXiv:2107.11823
64. Li X-Y, Lei W-J, Yang Y-B (2022) From easy to hard: two-stage selector and reader for multi-hop question answering. CoRR. arXiv:2205.11729
65. Guan Y, Li Z, Leng J et al (2021) Block-skim: efficient question answering for transformer. CoRR. arXiv:2112.08560
66. Zhou X, Hu B, Chen Q, Wang X (2018) Recurrent convolutional neural network for answer selection in community question answering. Neurocomputing 274:8–18. <https://doi.org/10.1016/j.neucom.2016.07.082>
67. Cohen D, Yang L, Croft WB (2018) WikiPassageQA: a benchmark collection for research on non-factoid answer passage retrieval. In: 41st international ACM SIGIR conference on research and development in information retrieval, SIGIR 2018. Association for Computing Machinery, Inc, pp 1165–1168
68. Zhang X, Li S, Sha L, Wang H (2017) Attentive interactive neural networks for answer selection in community question answering. In: Proceedings of the AAAI conference on artificial intelligence, vol 31, no 1
69. Bian W, Li S, Yang Z et al (2017) A compare-aggregate model with dynamic-clip attention for answer selection. In: International conference on information and knowledge management, proceedings. Association for Computing Machinery, pp 1987–1990
70. Yoon S, Derroncourt F, Kim DS et al (2019) A compare-aggregate model with latent clustering for answer selection. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 2093–2096
71. Peng Y, Liu B (2018) Attention-based neural network for short-text question answering. In: ACM International conference proceeding series. Association for Computing Machinery, pp 21–26
72. Yu AW, Dohan D, Luong M-T et al (2018) QANet: combining local convolution with global self-attention for reading comprehension. CoRR. arXiv:1804.09541
73. Miller A, Fisch A, Dodge J et al (2016) Key-value memory networks for directly reading documents. In: Proceedings of the 2016 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1400–1409
74. Yang L, Ai Q, Guo J, Croft WB (2016) aNMM: ranking short answer texts with attention-based neural matching model. In: International conference on information and knowledge management, proceedings. Association for Computing Machinery, pp 287–296
75. Shao T, Guo Y, Chen H, Hao Z (2019) Transformer-based neural network for answer selection in question answering. IEEE Access 7:26146–26156. <https://doi.org/10.1109/ACCESS.2019.2900753>
76. Sukhbaatar S, Szlam A, Weston J, Fergus R (2015) End-to-end memory networks. In: Advances in neural information processing systems, pp 2440–2448
77. Kumar A, Irsoy O, Ondruska P et al (2016) Ask me anything: dynamic memory networks for natural language processing. In: International conference on machine learning, pp 1378–1387
78. Pan B, Li H, Zhao Z et al (2017) MEMEN: multi-layer embedding with memory networks for machine comprehension. In: AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18)

79. Back S, Yu S, Indurthi SR et al (2018) MemoReader: large-scale reading comprehension through neural memory controller. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 2131–2140
80. Xiong C, Zhong V, Socher R (2016) Dynamic coattention networks for question answering. In: International conference on learning representations (ICLR)
81. Wang S, Yu M, Chang S, Jiang J (2018) A co-matching model for multi-choice reading comprehension. In: Association for computational linguistics (ACL), pp 746–751
82. Xiong C, Zhong V, Socher R (2017) DCN+: mixed objective and deep residual coattention for question answering. CoRR. [arXiv:1711.00106](https://arxiv.org/abs/1711.00106)
83. McCann B, Keskar NS, Xiong C, Socher R (2018) The Natural language decathlon: multitask learning as question answering. CoRR. [arXiv:1806.08730](https://arxiv.org/abs/1806.08730)
84. Wang W, Yan M, Wu C (2018) Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1: long papers, pp 1705–1714
85. Tay Y, Tuan LA, Hui SC (2018) Multi-cast attention networks. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY, USA, pp 2299–2308
86. Min S, Seo M, Hajishirzi H (2017) Question answering through transfer learning from large fine-grained supervision Data. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 2: short papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 510–517
87. Golub D, Huang P-S, He X, Deng L (2017) Two-stage synthesis networks for transfer learning in machine comprehension. In: Proceedings of the 2017 conference on empirical methods in natural language processing. association for computational linguistics, Stroudsburg, PA, USA, pp 835–844
88. Seo M, Kembhavi A, Farhadi A, Hajishirzi H (2016) Bidirectional attention flow for machine comprehension. In: International conference on learning representations (ICLR)
89. Liu X, Shen Y, Duh K, Gao J (2018) Stochastic answer networks for machine reading comprehension. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1694–1704
90. Xiong W, Yu M, Guo X et al (2019) Simple yet effective bridge reasoning for open-domain multi-hop question answering. In: Proceedings of the 2nd workshop on machine reading for question answering. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 48–52
91. Hermann KM, Kočiský T, Grefenstette E et al (2015) Teaching machines to read and comprehend. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol 28. Curran Associates Inc, Red Hook, NY
92. Kadlec R, Schmid M, Bajgar O, Kleindienst J (2016) Text understanding with the attention sum reader network. In: Proceedings of the 54th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 908–918
93. Trischler A, Ye Z, Yuan X et al (2016) Natural language comprehension with the EpiReader. In: Proceedings of the 2016 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 128–137
94. Wang S, Jiang J (2017) Machine comprehension using match-LSTM and answer pointer. In: International conference on learning representations (ICLR), pp 1–15
95. Cui Y, Chen Z, Wei S et al (2017) Attention-over-attention neural networks for reading comprehension. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 593–602
96. Dhingra B, Liu H, Yang Z et al (2017) Gated-attention readers for text comprehension. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 1832–1846
97. Wang W, Yang N, Wei F et al (2017) Gated self-matching networks for reading comprehension and question answering. In: ACL 2017—55th annual meeting of the association for computational linguistics, proceedings of the conference (long papers). Association for Computational Linguistics (ACL), pp 189–198
98. Liu R, Wei W, Mao W, Chikina M (2017) Phase conductor on multi-layered attentions for machine comprehension. CoRR. [arXiv:1710.10504](https://arxiv.org/abs/1710.10504)
99. Huang H-Y, Zhu C, Shen Y, Chen W (2017) FusionNet: fusing via fully-aware attention with application to machine comprehension. CoRR. [arXiv:1711.07341](https://arxiv.org/abs/1711.07341)
100. Zhu H, Wei F, Qin B, Liu T (2018) Hierarchical attention flow for multiple-choice reading comprehension. In: Hierarchical Attention flow for multiple-choice reading comprehension, vol 32, no 1

101. Kundu S, Ng HT (2018) A question-focused multi-factor attention network for question answering. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, no 1
102. Tan C, Wei F, Yang N et al (2018) S-Net: from answer extraction to answer synthesis for machine reading comprehension. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, no 1
103. Zhu C, Zeng M, Huang X (2018) SDNet: contextualized attention-based deep network for conversational question answering. CoRR. [arXiv:1812.03593](https://arxiv.org/abs/1812.03593)
104. LeeKim HH (2020) GF-Net: improving machine reading comprehension with feature gates. Pattern Recognit Lett 129:8–15. <https://doi.org/10.1016/j.patrec.2019.10.030>
105. Huang X, Zhang J, Li D, Li P (2019) Knowledge graph embedding based question answering. In: WSDM 2019—proceedings of the 12th ACM international conference on web search and data mining. Association for Computing Machinery, Inc, pp 105–113
106. Chen Y, Wu L, Zaki MJ (2020) GraphFlow: exploiting conversation flow with graph neural networks for conversational machine comprehension. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence. International Joint Conferences on Artificial Intelligence Organization, California, pp 1230–1236
107. Cao X, Liu Y (2022) Coarse-grained decomposition and fine-grained interaction for multi-hop question answering. J Intell Inf Syst 58:21–41. <https://doi.org/10.1007/s10844-021-00645-w>
108. Nishida K, Nishida K, Nagata M et al (2019) Answering while summarizing: multi-task learning for multi-hop QA with evidence extraction. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 2335–2345
109. Xiao Y, Qu Y, Qiu L et al (2019) dynamically fused graph network for multi-hop reasoning. In: Proceedings of the 57th annual meeting of the association for computational linguistics, Florence, Italy, pp 6140–6150
110. Cao Y, Fang M, Tao D (2019) BAG: bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): human language technologies, pp 357–362
111. Fang Y, Sun S, Gan Z et al (2020) Hierarchical graph network for multi-hop question answering. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 8823–8838
112. Zhang M, Li F, Wang Y et al (2020) Coarse and fine granularity graph reasoning for interpretable multi-hop question answering. IEEE Access 8:56755–56765. <https://doi.org/10.1109/ACCESS.2020.2981134>
113. Tu M, Huang K, Wang G et al (2020) Select, answer and explain: interpretable multi-hop reading comprehension over multiple documents. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, no 05, pp 9073–9080
114. Zhao C, Xiong C, Rosset C et al (2020) Transformer-XH: multi-evidence reasoning with extra hop attention. In: International conference on learning representations (ICLR)
115. Zhang X, Bosselut A, Yasunaga M et al (2022) GreaseLM: Graph REASONing Enhanced Language Models for question answering. CoRR. [arXiv:2201.08860](https://arxiv.org/abs/2201.08860)
116. Shi J, Cao S, Hou L et al (2021) TransferNet: an effective and transparent framework for multi-hop question answering over relation graph. In: Proceedings of the 2021 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 4149–4158
117. Zhang Y, Nie P, Ramamurthy A, Song L (2021) Answering any-hop open-domain questions with iterative document reranking. In: SIGIR 2021—proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. Association for Computing Machinery, Inc, pp 481–490
118. Ren H, Dai H, Dai B et al (2021) LEGO: latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In: International conference on machine learning, pp 8959–8970
119. Xiong W, Li XL, Iyer S et al (2020) Answering complex open-domain questions with multi-hop dense retrieval. In: Proceedings of the conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019, pp 2590–2602
120. Wu J, Mu T, Thiyyagalingam J, Goulermas JY (2020) Building interactive sentence-aware representation based on generative language model for community question answering. Neurocomputing 389:93–107. <https://doi.org/10.1016/j.neucom.2019.12.107>
121. Bi B, Wu C, Yan M et al (2019) Incorporating external knowledge into machine reading for generative question answering. In: Conference on empirical methods in natural language processing and international joint conference on natural language processing (EMNLP-IJCNLP)

122. Bauer L, Wang Y, Bansal M (2018) Commonsense for generative multi-hop question answering tasks. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 4220–4230
123. Izacard G, Grave E (2021) Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 874–880
124. Yavuz S, Hashimoto K, Zhou Y et al (2022) Modeling multi-hop question answering as single sequence prediction. In: Proceedings of the 60th annual meeting of the association for computational linguistics, vol 1: long papers. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 974–990
125. Shen Y, Huang P sen, Gao J, Chen W (2017) ReasoNet: learning to stop reading in machine comprehension. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, pp 1047–1055
126. Buck C, Bulian J, Ciarmita M et al (2018) Ask the right questions: active question reformulation with reinforcement learning. In: International conference on learning representations (ICLR)
127. Xu Y, Liu J, Gao J et al (2017) Dynamic fusion networks for machine reading comprehension. CoRR. [arXiv:1711.04964](https://arxiv.org/abs/1711.04964)
128. Hu M, Peng Y, Huang Z et al (2018) Reinforced mnemonic reader for machine reading comprehension. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence. International Joint Conferences on Artificial Intelligence Organization, California, pp 4099–4106
129. Santoro A, Raposo D, Barrett DGT et al (2017) A simple neural network module for relational reasoning. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates Inc, Red Hook, NY
130. Swayamdipta S, Parikh AP, Kwiatkowski T (2017) Multi-mention learning for reading comprehension with neural cascades. In: International conference on learning representations (ICLR)
131. Tay Y, Tuan LA, Hui SC (2018) Hyperbolic representation learning for fast and efficient neural question answering. In: WSDM 2018—proceedings of the 11th ACM international conference on web search and data mining. Association for Computing Machinery, Inc, pp 583–591
132. Seonwoo Y, Kim J-H, Ha J-W, Oh A (2020) Context-aware answer extraction in question answering. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 2418–2428
133. Wu Y, Zhao S (2021) Community answer generation based on knowledge graph. Inf Sci 545:132–152. <https://doi.org/10.1016/j.ins.2020.07.077>
134. Zhou G, Xie Z, Yu Z, Huang JX (2021) DFM: a parameter-shared deep fused model for knowledge base question answering. Inf Sci 547:103–118. <https://doi.org/10.1016/j.ins.2020.08.037>
135. He H, Gimpel K, Lin J (2015) Multi-perspective sentence similarity modeling with convolutional neural networks. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Association for Computational Linguistics, pp 1576–1586
136. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
137. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681. <https://doi.org/10.1109/78.650093>
138. Wang S, Jiang J (2017) A compare-aggregate model for matching text sequences. In: Proceedings of the 5th international conference on learning representations (ICLR 2017)
139. Vinyals O, Fortunato M, Jaitly N (2015) Pointer networks. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol 28. Curran Associates Inc, Red Hook, NY
140. Gong Y, Bowman SR (2018) Ruminating reader: reasoning with gated multi-hop attention. In: Proceedings of the workshop on machine reading for question answering. Association for Computational Linguistics, pp 1–11
141. Liu H, Singh P (2004) ConceptNet—a practical commonsense reasoning tool-kit. BT Technol J 22:211–226. <https://doi.org/10.1023/B:BTJ.0000047600.45421.6d>
142. Yang Y, Yih W-T, Meek C (2015) WIKIQA: a challenge dataset for open-domain question answering. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 2013–2018
143. Filice S, Croce D, Moschitti A, Basili R (2016) KeLP at SemEval-2016 task 3: learning semantic relations between questions and answers. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 1116–1123
144. Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 2383–2392

145. Weston J, Bordes A, Chopra S et al (2015) Towards AI-complete question answering: a set of prerequisite toy tasks. CoRR. [arXiv:1502.05698](https://arxiv.org/abs/1502.05698)
146. Lai G, Xie Q, Liu H et al (2017) RACE: large-scale ReAding Comprehension Dataset From Examinations. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 785–794
147. Zhang S, Liu X, Liu J et al (2018) ReCoRD: bridging the gap between human and machine commonsense reading comprehension. CoRR. [arXiv:1810.12885](https://arxiv.org/abs/1810.12885)
148. Joshi M, Choi E, Weld DS, Zettlemoyer L (2017) TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1: long papers
149. Kočiský T, Schwarz J, Blunsom P et al (2018) The NarrativeQA reading comprehension challenge. Trans Assoc Comput Linguist 6:317–328. https://doi.org/10.1162/tacl_a_00023
150. Nguyen T, Rosenberg M, Song X et al (2016) MS MARCO: a human generated machine reading comprehension dataset. In: CoCo@ NIPS
151. Yang Z, Qi P, Zhang S et al (2018) HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics
152. Wang S, Yu M, Chang S, Jiang J (2018) A co-matching model for multi-choice reading comprehension. arXiv preprint. [arXiv:1806.04068](https://arxiv.org/abs/1806.04068)
153. Choi E, He H, Iyyer M et al (2018) QuAC: question answering in context. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics Brussels, Belgium, pp 2174–2184
154. Brown T, Mann B, Ryder N et al (2020) Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R et al (eds) Advances in neural information processing systems. Curran Associates Inc, Red Hook, NY, pp 1877–1901
155. Smith S, Patwary M, Norick B et al (2022) Using DeepSpeed and megatron to train megatron-turing NLG 530B, a large-scale generative language model. CoRR. [arXiv:2201.11990](https://arxiv.org/abs/2201.11990)
156. Ahmed W, Antó BP (2017) Question answering system based on neural networks. Int J Eng Res 6:142–144
157. Romeo S, da San MG, Belinkov Y et al (2019) Language processing and learning models for community question answering in Arabic. Inf Process Manage 56:274–290. <https://doi.org/10.1016/j.ipm.2017.07.003>
158. Aouichat A, Hadj Ameur MS, Geussoum A (2018) Arabic question classification using support vector machines and convolutional neural networks. In: International conference on applications of natural language to information systems, pp 113–125
159. Park C, Lee C, Hong L et al (2019) S2-Net: machine reading comprehension with SRU-based self-matching networks. ETRI J 41:371–382. <https://doi.org/10.4218/etrij.2017-0279>
160. Liu J, Yang Y, Lv S et al (2019) Attention-based BiGRU-CNN for Chinese question classification. J Ambient Intell Humaniz Comput. <https://doi.org/10.1007/s12652-019-01344-9>
161. Lai Y, Feng Y, Yu X et al (2019) Lattice CNNs for matching based Chinese question answering. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, no 01, pp 6634–6641

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Heba Abdel-Nabi is a Ph.D. student in Computer Science at Princess Sumaya University for Technology (PSUT). She received her Bachelor's degree in Computer Engineering in 2010 and her Master's degree in Electrical Engineering in 2015 from PSUT. Her research interests are digital image processing and information security, deep learning, artificial intelligence, and evolutionary algorithms.



Prof. Arafat Awajan is a full professor of computer science; he received his Ph.D. degree in Computer Science from the University of Franche-Comte, France in 1987. He held different academic positions at the Royal Scientific Society and Princess Sumaya University for Technology and Mutah University. He is currently the president of Mutah university in Jordan. His research interests include: Natural Language Processing, Arabic Text Mining and Digital Image Processing.



Mostafa Z. Ali received the Bachelor degree in Applied Mathematics at Jordan University of Science & Technology (JUST), Irbid, Jordan, in 2000. He finished his Masters in Computer Science at the University of Michigan-Dearborn, Michigan, USA in 2003. He finished his Ph.D. in computer science/Artificial Intelligence at Wayne State University, Michigan, USA in 2008. He is a professor at the department of computer information systems at Jordan University of Science & Technology, Irbid, Jordan. He is an associate editor of the Swarm and Evolutionary Computation (SWEVO), an Elsevier journal, and Information Sciences (INS), an Elsevier journal. His research interests include Artificial Intelligence applications, evolutionary computation, Machine Learning, Deep Learning, Virtual/Augmented Reality, and gaming. Dr. Ali is a member of the IEEE, the IEEE computer society, the American Association of Artificial Intelligence (AAAI), and the ACM.