Neural Computing and Applications (2022) 34:2765–2783
https://doi.org/10.1007/s00521-021-06748-3

REVIEW

Recent progress in leveraging deep learning methods for question answering

Tianyong Hao[1] · Xinxin Li[2] · Yulan He[1] · Fu Lee Wang[3] · Yingying Qu[4]

Received: 21 August 2020 / Accepted: 11 November 2021 / Published online: 16 January 2022

Abstract

Question answering, serving as one of important tasks in natural language processing, enables machines to understand questions in natural language and answer the questions concisely. From web search to expert systems, question answering systems are widely applied to various domains in assisting information seeking. Deep learning methods have boosted various tasks of question answering and have demonstrated dramatic effects in performance improvement for essential steps of question answering. Thus, leveraging deep learning methods for question answering has drawn much attention from both academia and industry in recent years. This paper provides a systematic review of the recent development of deep learning methods for question answering. The survey covers the scope including methods, datasets, and applications. The methods are discussed in terms of network structure characteristics, methodology innovations, and their effectiveness. The survey is expected to be a contribution to the summarization of recent research progress and future directions of deep learning methods for question answering.

Keywords Question answering · Deep learning · Methods · Dataset · Performance evaluation

1 Introduction

Question answering (QA) aims at precisely answering a given question in natural language. Instead of requiring a formatted query and returning a list of relevant documents, QA enables human users to interact with machines in a more natural way. It is a classical research problem in natural language processing (NLP), involving techniques of question analysis, answer retrieval, and answer ranking [33]. QA has been studied for many years and implemented in various types of systems. The Baseball [29] is regarded as the earliest QA system, while more recent chatbots such as Siri, Alexa, and Cortana are popular. However, from manually constructed text template in earlier years [100] to very recent technologies, there still remains plenty of challenges in QA and many studies have been dedicated to improve QA systems.

Deep learning (DL), as a class of machine learning methods involving deep architecture, has drawn a lot of attention in recent years and has been verified in various domains. The common paradigm of classical machine learning consists of feature engineering and a machine learning algorithm, which requires a large effort to build

✉ Yingying Qu
jessie.qu@gdufs.edu.cn

Tianyong Hao
haoty@m.scnu.edu.cn

Xinxin Li
lixx233@mail2.sysu.edu.cn

Yulan He
jasminehyl@nyu.edu

Fu Lee Wang
pwang@hkmu.edu.hk

[1] School of Computer Science, South China Normal University, Guangzhou, China

[2] Institute of Logic and Cognition, Sun Yat-sen University, Guangzhou, China

[3] School of Science and Technology, Hong Kong Metropolitan University, Hong Kong, Hong Kong SAR

[4] School of Business, Guangdong University of Foreign Studies, Guangzhou, China

hand-crafted features [18]. Largely developed since 2006 [2], deep learning provides an elegant way of machine learning by using plenty of training data. Applying deep learning in an end-to-end fashion can free researchers from tedious feature engineering [51]. Moreover, DL methods are able to uncover hidden patterns regarding different tasks when training with the downstream tasks. At the same time, DL methods are versatile [7]. The same framework can be adopted in different tasks and yield reasonable results. Therefore, the innovations in DL methods are not limited to specific applications or domains.

The network architectures of deep learning methods consist of multiple layers, which include plenty of non-linear processing units and thus allow the methods to capture hierarchical features and concepts [2, 51] as low-dimensional vectors [102] and then perform downstream tasks. Deep learning methods are widely applied in various NLP tasks, such as text classification [60], named entity recognition [53] and word embedding [88].

In recent works, Dimitrakis et al. [20] addressed a detailed discussion on the past surveys on QA area up to 2018, and reviewed QA systems with linked data and documents. Shah et al. [72] analyzed the methods under web-based QA systems and presented a comparison on methods and techniques of web-based QA systems with the same datasets and evaluation metrics. Ojokoh and Adebisi [61] investigated the generic QA frameworks and addressed current challenges.

Over the past few years, an increasing number of studies on deep learning methods have been proposed to improve performance of various QA tasks. Plenty of these methods successfully achieve state-of-the-art performance. The ideas behind these methods are often inspiring and transferable to other applications. However, there is a lack of detailed survey on recent deep learning architectures adopted in QA tasks. Moreover, different QA frameworks may vary a lot, which largely depends on knowledge sources, question types, and expected answer types. The underlying knowledge sources of QA can be unstructured plain text, knowledge graph, or question–answer pairs [43]. The expected answers can be a paragraph, a description, or an option letter. Hence, required techniques of building QA systems are simplified and divided into different QA tasks, where works on each kind of QA task are often based on similar predefined premises, being investigated and validated upon several popular datasets. As a result, in order to gain a deeper insight into these tasks, one should not only be familiar with methods but also with the adopted QA tasks and datasets in each domain.

In this paper, we summarize the current research status, which is expected to be a contribution to the analysis of recent deep learning methods for QA. Meanwhile, we discuss the recent works regarding different tasks in QA area, including question classification, answer extraction, question–answer matching, knowledge base question answering, and question generation. The difference between previous surveys and ours is that we focus on the most recent deep learning methods covering multiply QA tasks. In addition, innovations and advantages of network structures adopted in the methods are discussed.

The rest of the paper is organized as follows: Section 2 to Section 6 introduces the QA systems, and reviews the QA methods and techniques, covering the recent literature based on the objectives of QA, i.e., question classification, answer extraction, question–answer matching, knowledge base question answering, and question generation. Section 7 describes evaluation metrics and presents a list of evaluation datasets. Section 8 summarizes typical deep learning methods in QA, provides a comparison of different models' performances on popular datasets, and discusses the current main challenges and trends of deep learning-based QA. Section 9 concludes the paper.

2 Question classification

Question classification, aiming to determine the target types of posted questions or expected answer types of questions [16], plays an important role in QA systems. Efficient question classification can promote the accuracy of QA systems by determining answer types and reducing search scope [72].

The capability of convolutional neural network (CNN) to extract different scales of features make it effective in classification tasks. Banerjee et al. [4] presented a framework combining feature engineering with CNNs for Bengali-English code mixed cross script factoid question classification, which involved different forms of user-generated noise, such as different contracted words, omitted punctuations, common vocabulary in different language, and so on. The feature vectors obtained after feature engineering were concatenated into the vector representation of questions. Then, a feature map was generated by a convolution operation, to which a max pooling operation was further applied in order to capture the most important features. The proposed model used multiple filters with various window sizes, where each filter extracted only one feature. Finally, the obtained features were fed into a dense layer and an output layer to acquire classification result. Aiming to improve short-text representation in convolution, Zhang et al. [108] proposed a dependency-based CNN (depCNN) model. By integrating a dependency layer into CNNs, the model overcame the shortcoming of CNNs on insufficiently capture syntactic information inside a sentence. Specifically, depCNN used a dependency layer to map the depths of words in dependency syntax trees into

continuous real space instead of directly taking concatenated word vectors as input. Thus, the mapping results could be used to reweight word vectors which allowed more compositional feature maps. Thereafter, CNNs extracted features from weighted word vectors for further purposes. Besides, depCNN was extended to learn the interactions between text pairs. They also applied depCNN on a text classification task, duplicate classification task, and text pair ranking task. Relying on dependency parsing, depCNN had some extent of robustness since dependency parsing introduced some noise. The related experiments involved different types of questions and texts in Chinese and in English.

Liu et al. [57] performed question classification task for user intent classification combining BERT [19] with Capsule network, incorporating a focal loss. Motivated by the effectiveness of introducing the focal loss in image object detection [55], taking the imbalance of dataset into consideration, the focal loss was taken in, replacing the cross-entropy loss function. Input embeddings contained token information, position information, and segment information were fed into stacked transformers encoder which employed pre-trained language model BERT. Capsule networks with dynamic routing mechanism were utilized to extract features from obtained encoded sequence. The classification result was produced by dense operation and softmax function. Experiments were conducted upon SNIPS, StackOverFlow, ECDT, and FDQuestion datasets, containing English and Chinese questions, covering different intent categories and various topics.

Croce et al. [17] combined kernel methods and neural networks into a novel architecture called Kernel-based Deep Architecture (KDA). They used the Nyström method to provide an approximation of kernel functions and adapted it to feed a neural network. The experiments of applying in question classification were conducted on UIUC datset, which contained various types of questions. Each question was projected into Nyström space and the resulting embeddings were grouped into clusters corresponding to 6 gold question classes.

We summarize the DL structures, datasets, and evaluation metrics adopted in the methods for question classification task as Table 1.

3 Answer extraction

Answer extraction provides exact answers from related documents or statements. It usually involves candidate answer extraction and answers ranking processes [61]. However, this task is challenging since question facts are usually short and limited when involving multiple entities. Building an end-to-end framework incorporating deep learning architectures enables the models to produce expected answers without taking separate processes.

Jing et al. [39] proposed a new architecture Gated Orthogonal Recurrent Unit (GORU), and applied it in an end-to-end style. The RNNs with GORU were a novel RNNs-based model that combined the remembering ability of unitary RNNs with the ability of gated RNNs to effectively forget redundant or irrelevant information in memory. To apply the structure in QA tasks, an end-to-end architecture was adopted. Two separate RNNs were fed with word embeddings of statements and questions. Then, the outputs were concatenated and fed into the final RNN for answer generation. The proposed model was tested on the bAbI dataset to verify the ability of RNNs on language understanding and basic logical reasoning, where the model was required to answer a short question based on several sentence descriptions.

End-to-end memory network (MemN2N) proposed by Sukhbaatar et al. [79] has shown its effectiveness in different research field. MemN2N is trained end to end, and thus less supervision is required in training phase. At the same time, the external memory represents the context for inference. However, MemN2N failed on smaller datasets or the case requiring a larger memory. Hence, introducing multiscale notions of attention may be promising. Wulamu et al. [93] proposed two improved methods, a novel Gated Linear Units (GLU) and local-attention-based end-to-end memory networks (MemN2N-GL), based on MemN2N for QA tasks on the bAbI dataset. The proposed methods were expected to extract more useful interactions between memory and questions. First, they introduced a local-attention mechanism into MemN2N in correlation calculation. Then, the GLU was used to optimize the updating of hidden state between the layers of MemN2N. Experiment results demonstrated that MemN2N-GL outperformed the original MemN2N. To perform the QA tasks on the bAbI dataset using MemN2N in an easier way, Yang and Fan [98] proposed a convolutional end-to-end memory network (CMemN2N), which was a combination of convolutional architecture and end-to-end memory networks . CMemN2N used convolution computation to generate sentence vector representations. Incorporating the memory network architecture, CMemN2N could efficiently capture the clues needed for reasoning process by abstracting local information in contexts and questions.

Dynamic memory networks (DMN) [44] had shown its potential with an iterative attention mechanism. Yue et al. [104] introduced a method to extract global and hierarchical salient features from input questions at the same time, and utilized these features to construct multiple feature sets. The proposed model, Enhanced Question Understanding with Dynamic Memory Networks (EnDMN), was based on dynamic memory networks. In the

Table 1 The DL structures, datasets, and evaluation metrics adopted in the methods for question classification task

Work	Network Structures	Datasets	Metrics
Banerjee et al. [4]	CNNs	MSIR16 Datasets	Precision, Recall, F1
depCNN [108]	CNNs	TREC, MR, CR, MPQA, SST2, SST5, Clinic-1	Accuracy, F1, MAP, MRR
BERT-Cap [57]	BERT and Capsule network	SNIPS, StackOverFlow, ECDT, and FDQuestion	Precision, Recall, F1, and Accuracy
KDA [17]	The kernel methods and neural networks	UIUC, FrameNet dataset (1.3 version), SemEval-2016 task 3 challenge	Accuracy, F1

EnDMN, a global representation of questions was produced by a gated recurrent neural network. A salient representation of questions was generated by another question module which contained a gated recurrent neural network and a max-pooling layer. The global representation was expected to include common knowledge of questions, and the salient representation was supposed to extract salient features from the questions. Experiment results on the bAbI dataset showed that this modification in the question module promoted the model performance in answer extraction.

To enable models to capture variability of natural language and yield diverse replies, Parshakova et al. [64] proposed a module called Adapts Parameters through Interpretation Policy (APIP), which was integrated within a Document Retriever Question Answering (DrQA) model. A training framework was designed to learn a complex distribution of latent question interpretations simultaneously with a QA training procedure. A discrete interpretation variable was used to make the Simple Recurrent Unit (SRU) weights to adapt to a particular interpretation, which allowed the model to implement several answering modes. They proved that updating the latent distribution with rewards from a variational lower bound provided an effective learning approach. A new way to integrate the variational inference framework within a QA model was introduced by parameter adaptation. The experiments were conducted on SQuAD 1.1, which contained longer descriptions and various question types, when questions had several ground truth.

In answer sentences selection task, instead of matching a question and individual candidate sentence separately, Tan et al. [83] modeled context information with hierarchical gated recurrent neural networks and introduced the gate mechanism to select matching information between sentences and queries at both word level and sentence level. The framework was based on Bi-RNN with GRUs, incorporating context independent matching and context dependent matching. The experiments were conducted on two answer sentence selection benchmark datasets, WikiQA and SQuAD.

To extract answers for a multi-passage task, an extract-then-select framework was introduced by Wang et al. [89]. In this framework, an extractor generated answer candidates, and then a selector decided a final answer with information over all answer candidates. Following the structure of the framework, Ren et al. [68] adopted generative adversarial training to utilize unlabeled passages for model training. Extractor was trained in multi-task learning way, where additional tasks used generative adversarial training to train the extractor with a discriminator. To enable the backpropagation in generative training process, a hybrid method for predicting answer candidates was proposed by combining boundary-based and content-based methods. Quasar-T, SearchQA, and TriviaQA-unfiltered were the datasets applied in experiments.

Cai et al. [11] proposed a stacked Bi-LSTM model with coattention mechanism to extract the interactions between questions and answers. In the coattention mechanism, the output vectors of Bi-LSTMs were used to calculate an affinity matrix L, which included affinity scores corresponding of all pairs of question and answer words. A softmax was applied on L to compute attention weights, which were concatenated with affinity matrix to compute new context vectors. For answer representation, an attentive attention mechanism was applied. A max pooling was taken to convert the output of coattention into a fixed-length vector output O. The final representation was the weighted sum of context vectors and O, where the weights were learned via the attention mechanism. Finally, the similarity between the vector representations of the questions and the answers was calculated by a function combining a cosine similarity and a Euclidean distance. Evaluated on TREC 8-13 dataset and Wiki-QA dataset, experiment results showed the effectiveness of the proposed coattention mechanism between questions and answers. In Cai et al. [12], this model was modified and applied in restricted-domain question answering. Compared with previous frameworks, a CNN module was

added before the stacked Bi-LSTM to improve analysis and prediction.

Song et al. [77] introduced a positional CNN (P-CNN) model embedding positional information into each layer of the model to enhance text matching. Considering the difficulty of CNNs to capture locations of aligned words and influence of aligned words in text pair, P-CNN aggregated positional information from multiple perspectives, including word level, phrase level, and sentence level. In the position-similarity mapping layer, a position-similarity matrix was generated to project word-level positional information into local matching signals. In the convolution layers, a position sensible convolution filter was designed to model various levels of positional information for effective detection and extraction of positional information. Thereafter, local matching signals at the three levels were aggregated to a matching function in the fully connected layer. Finally, in the multiple-perspective aggregating layer, they incorporated positional information from multiple perspectives and generated the final matching scores for question and document pairs. Through experiments on WikiQA, they found that the sentence-level positional information played the most important role in boosting the performance of P-CNN. Multiple-perspective positional information was effective in boosting the neural matching performance for web search. And interaction-focused models performed better experimentally than those representation-focused models.

To investigate the potential benefit of attention mechanism, Huang et al. [35] introduced an attention mechanism into the sentence feature level. They proposed a Question-oriented Feature Attention (QFA) mechanism and applied it to the tasks of machine reading comprehension, answer selection, and community-based question answering. A QFA-based reading comprehension model was proposed for machine reading comprehension. After extracting the features from a question and its answers, the question was fed into RNNs to generate attention weights for each extracted feature. The weighted features were fed into the answer selection models for downstream tasks. The model was tested on MCTest, a reading comprehension dataset with a set of texts and associated multiple-choice questions, each with four answer candidates. For the machine comprehension task, Lee et al. [52] proposed a model combining a Tree-LSTM and an attention mechanism. The proposed hierarchical attention model fed word sequences of questions, answers, and stories into the Tree-LSTM to generate sentence representations. The attention mechanism was used to compute question related attention weights for different sentences. Experiment results demonstrated that applying the attention mechanism over the story could extract better representations.

We summarize the DL structures, datasets, and evaluation metrics adopted in the models for answer extraction task as Table 2.

4 Question–Answer matching

In this section, we focus on studies on question–answer matching, including community question answering, question–answer pair matching, and answer selection. Among these studies, a typical paradigm is to project questions and answers into a consistent vector space where similarity can be measured.

In recent years, community question answering (cQA) has become a hot topic due to the flourish of question–answering online communities, such as StackOverflow and Quora. Based on history question answering records, much effort has been made to find similar questions or fetch exact answers. To address the issue, the measurements of question similarity and question–answer similarity are required. The task involves candidates retrieval and candidates ranking, which can be simplified as an answer classification or an answer ranking problem.

A Kernel-based Deep Architecture (KDA) was proposed by Croce et al. [17], which incorporates the kernel methods and neural networks into a novel architecture. KDA was applied on cQA task in SemEval-2016 task-3: challenge subtask A. The subtask A was a binary classification problem and a ranking problem to perform re-ranking comments based on their utility in answering questions. The proposed model performed better than other kernel methods in experiments containing question-comment pairs as data instances.

Yang et al. [97] introduced a model called multi-head interactive attention network for community question answering in a binary classification form. To enhance the performance, external common sense knowledge from a knowledge base was incorporated to capture entities and relations. In order to learn question representations, they leveraged question categorization models to locate salient information of questions more accurately. A multi-head interactive attention mechanism was applied to reduce the influence of redundant and noisy information. The experiments on SemEval-2015, 2016, and 2017 datasets showed that the model achieved the best performance compared with baseline methods.

Ben Abacha and Demner-Fushman [6] developed a QA system on the basis of Recognizing Question Entailment (RQE) to match similar questions, and experiments were performed on three datasets including Quora, Clinical-QE, and SemEval 2016 Task 3B. First, they compared logistic regression (LR) and DL methods for RQE on different datasets. The results showed that the DL model performed

Table 2 The DL structures, datasets, and evaluation metrics adopted in the methods for answer extraction task

Work	Network structures	Datasets	Metrics
GORU [39]	RNNs	bAbI dataset	Accuracy
GLU and MemN2N-GL [93]	MemN2N	bAbI dataset	Error Rates (%)
CMemN2N [98]	Convolutional architecture and the end-to-end memory networks	bAbI dataset	Average Accuracy
DrQA [64]	An APIP module	SQuAD 1.1	F1, EM
EnDMN [104]	Dynamic memory networks (DMN)	bAbI dataset	Mean and Minimum Error Rates (%)
Tan et al. [83]	Hierarchical gated recurrent neural networks	WikiQA, SQuAD	MAP, MRR
QFA [35]	RNNs	MCTest, StackExchange	Accuracy, Precision@1 (P@1), MRR
HAM [52]	The Tree-LSTM	TOEFL listening comprehension test dataset	Accuracy
Ren et al. [68]	Bi-LSTM	Quasar-T, SearchQA, and TriviaQA-unfiltered	EM, F1
Cai et al. [11]	Stacked Bi-LSTM model	TREC 8-13 dataset, Wiki-QA	MAR, MRR
Cai et al. [12]	Stacked Bi-LSTM model and CNNs	CCKS2018, IPC-QA	Precision, Accuracy, Recall and F1
P-CNN [77]	CNNs	ClueWeb-09-Category-B1, TREC-QA and WikiQA	MAP, P@20 and nDCG@20, MAP and MRR

better when training and testing were conducted on the same dataset. In the entailment-based QA system, candidate questions were fetched using information retrieval models. The RQE method was applied to filter out non-entailed questions and re-rank remaining candidates. However, the DL method did not yield the best result in this framework. The hybrid method combining LR and IR provided the best results in the study.

Wen et al. [90] proposed a hybrid attention-based deep neural network called UIA-LSTM-CNN for answers selection in cQA. The hybrid attention mechanism combined local importance of a word in its current sentence and mutual importance of words in the counterpart sentence for sentence representation learning. In this way, the hybrid attention mechanism was able to find the most informative words for sentence matching. Besides, unlike most of the existing works, they did not only model semantic similarity between QA pairs but also model users with user-generated text to alleviate data sparsity problem. User representations were learned by explicitly attending over informative question parts. The hybrid attentive sentence matching and user modeling were integrated in an end-to-end unified neural network. The proposed model was validated on SemEval-2016 Task 3 and a Quora dataset, both containing user information of each question and answer.

Huang et al. [35] proposed a Question-oriented Feature Attention (QFA) mechanism and applied it to the task of machine reading comprehension, answer selection, and community-based question answering. A QFA-based reading comprehension model (QFAReader) was designed. It extracted features from questions and answers. After that, questions were fed into RNNs to generate attention weights for each extracted feature, which were further utilized in an answer selection model. In experiments conducted on self-crawled dataset, the aim was to rank the best answer at a higher position in the answer list of given questions.

Zhang et al. [106] used a character-level multi-scale CNNs architecture to learn the representation of Chinese medical text, in order to avoid the potential error propagation derived from Chinese word segmentation. In the proposed framework, convolutional multiple feature maps enabled the model to extract semantic information over different scales, and thus produced a better representation of questions and answers for question–answer matching. The proposed framework was validated on a newly created corpus named cMedQA, which provided Chinese medical questions with multiply answers. In experiments, the number of candidate answers was set to 100, including ground truth answers. For matching correct answers for Chinese medical questions but on a different dataset webMedQA, He et al. [32] built a Chinese medical QA framework that consists of Convolutional Semantic Clustered Representation (CSCR) and deep matching networks (MV-LSTM and MatchPyramid). The framework tried to address the Chinese word segmentation problem in medical

text by post-processing. In CSCR, character-level embeddings were concatenated into a sentence matrix with convolution operation. Through windowed max pooling, each row in the matrix was considered as the pattern of a word or clinical term. This work compared the performance of several CWS tools and conducted a serial of experiments on combinations of different input units and matching models.

In order to fully utilize the semantic information of sentences, Bi et al. [9] proposed a novel approach to apply attention mechanism on disease synonym detection in a Chinese medical QA system. The proposed model, called BLSTM-SFPA (the Bi-LSTM Model with Symptoms-Frequency Position Attention), utilized Ci-Lin to process symptom words in both questions and answers. This attention mechanism enabled the model to pay more attention to symptom words and their neighbors in description sentences.

Aiming to match relevant existing questions based on question sentence similarity, Chen et al. [13] proposed a framework called Heterogeneous Social Influential Network (HSIN) to integrate question textual content with asker social network information. The experiments were based on the Quora service and twitter user social network. The framework incorporated a random deep walk method with RNNs. Zhang et al. [109] introduced a new deep neural network-based sentence matching model, in which a sentence encoding structure called deep feature fusion was proposed to capture semantic eigenvalues by integrating multiple sequence encoding approaches. As for sentence matching task, the method was integrated into a deep learning architecture. In experiments on common Chinese semantic matching corpus LCQMC and public English semantic matching corpus Quora, the proposed framework had the promising performance. In the work of Mahmoud and Zrigui [59], a context-based approach was proposed for monolingual Arabic paraphrase detection to address the issue of semantic textual similarity detection in Arabic language. By using word2vec algorithm to encode documents, computational complexity and data sparsity problems were reduced. Sen2vec as a sentence vector representation was generated by the obtained averaged vectors. A CNN model was used with several statistic regularities to model documents and measure semantic similarity.

To predict whether a question would be answered or closed on cQA sites, Roy and Singh [69] used machine learning and deep learning models to predict closed questions and reasons why the questions are closed. Machine learning methods were used in question type classification with deep learning methods such as CNNs and LSTMs. Aiming to predict the reason for closed questions, five-class classification categories were built. Experiment results on the Stack Overflow dataset indicated that both machine learning methods and deep learning methods required a balanced dataset for better performance.

Adlouni et al. [1] compared the performance of various deep learning methods and machine learning methods on community question answering task in Arabic Language. The proposed models were evaluated on Semeval 2017 Task 3-Subtask D, where models determined relevance degree between queries and question–answer pairs.

We describe the DL structures, datasets, and evaluation metrics adopted in the models for question–answer matching task as Table 3.

5 Knowledge base question answering

Knowledge base question answering (KBQA), aiming to answer factoid questions with one or more knowledge bases (KBs) as the underlying data sources, is an important task in QA systems. Given a question, machine needs to fetch answers from KB based on question semantic or mentioned entities [20]. The basic idea of accomplishing querying knowledge base with natural language question is to obtain comparable representations of both question and underlying knowledge, involving techniques such as knowledge graph embedding, query generation and semantic parsing. Rather than directly obtaining target answers from KB knowledge, semantic parsing-based approaches construct query structures explicitly representing the semantic meaning of the questions. Incorporating deep learning techniques into semantic parsing is explored and is promising. In this section, the studies of KBQA are introduced.

To generate a query based on a natural language question, a popular pipeline is to detect and extract entities mentioned in the question, and then generate queries with linked entities. Sometimes multiply candidate queries are generated, where similarity measurement between questions and queries is required in order to select the best query. Zhu et al. [114] proposed a tree-to-sequence method for mapping natural language questions to executable queries. Firstly, candidate queries for a question from its linked entities were constructed. A LSTM-based model was employed to match queries against questions. During encoding, they proposed a tree-based LSTM to model contexts of an entity or relation in a query. The tree-based encoder encoded the structure of a candidate query. Then, a mixed-mode decoder which had two modes including generating mode and referring mode was used to select the best query. The generating mode focused on semantic-level correlation and the referring mode focused on surface-level correlations and the variations of language. Jiang et al. [38] implemented semantic querying of

Table 3 The DL structures, datasets, and evaluation metrics adopted in the methods for question–answer matching task

Work	Network structures	Datasets	Metrics
KDA [17]	The kernel methods and neural networks	UIUC, FrameNet, SemEval-2016 Task 3	Accuracy, F1
Zhang et al. [106]	CNNs	cMedQA	ACC@1
He et al. [32]	Convolutional Semantic Clustered Representation (CSCR) and deep matching networks (MV-LSTM and MatchPyramid)	webMedQA	Precision at 1 (P@1), MAP
UIA-LSTM-CNN [90]	LSTM, CNNs	SemEval-2016 Task 3, Quora	Accuracy, MAP
BLSTM-SFPA [9]	Bi-LSTM	MED-QA, GD-QA	MAP, MRR
QFA [35]	RNNs	MCTest, StackExchange	Accuracy, Precision@1 (P@1), MRR
HSIN [13]	RNNs	Collected from Quora and Twitter	MAP, MRR, Precision @N
Zhang et al. [109]	LSTM and FNNs	LCQMC, Quora	Precision, Recall, F1 and Accuracy
Mahmoud and Zrigui [59]	Word2vec, Sen2vec and CNNs	OSAC	F1
MKMIA-CQA [97]	RNNs and CNNs	SemEval-2015, SemEval-2016 and SemEval-2017	Accuracy, Macro-averaged F1, Recall, Precision

geographic knowledge in Chinese with a LSTM-CRF model and a multi-feature logistic model. To realize intelligent interaction with virtual geographical environments (VGEs) based on a geographic knowledge graph, questions expressed in natural language were processed using the model to identify geographic entities. The framework used a multi-feature logistic model to link geographic entities for resource mapping. Semantic combination was then performed to generate executable queries.

Instead of directly embed knowledge bases, a number of works transform subgraphs into word sequences, which can be embedded in the same way as questions. Lan et al. [47] converted entities and relations in knowledge bases into word sequences for question–answer matching. First, the word sequences were concatenated to form base candidate sequence. Then, contextual relations were applied to relation embedding with an attention mechanism. It was combined with base candidate sequences to form enhanced candidate sequences. A matching-aggregation framework was introduced to compute matching scores between candidate sequences and questions. The corresponding entity of the candidate sequence was selected as an answer.

To answer complex questions involving multiple predicates, Luo et al. [58] further extended the research in semantic parsing with neural network direction. The proposed method for semantic similarity measurement embedded the question and whole query graph into a uniform vector space. Query graph generation considered four kinds of constraints: entity, type, time, and ordinal constraints. It followed five steps: focus linking, main path generation, attaching entity constraints, type constraint generation, and time and ordinal constraint generation. After query graph generation, a query graph was split into predicate sequences, containing sequences of predicate ids and predicate names. The predicate sequences were embedded to obtain a semantic component vector. Question representation was produced by combining global representation and local representation. Global representation took token sequences as the input of Bi-GRUs, while local representation took dependency path as input of another Bi-GRUs. Before cosine similarity calculation, max pooling operations were applied to the vectors of semantic representation and question vectors. Additionally, an ensemble approach was proposed to enrich linking results from S-MART linker: building (mention, entity) pairs lexicon, constructing statistical features, and fitting linking scores with a 2-layer linear regression model. Detailed and comprehensive experiments were conducted to investigate the effectiveness of each module. This work encoded query graph as a whole rather than transform it into separate word sequences.

Based on deep learning techniques, an encoder-compare framework was adopted to obtain question representation

and knowledge base representation jointly. Hao et al. [31] proposed a strategy based on directed-acyclic-graph (DAG) embeddings to encode rich information conveyed by questions and knowledge bases. A joint learning framework based on RNNs and memory networks was presented for matching between textual questions and structural candidate answers. The framework consisted of four main modules, i.e., question encoder using Bi-LSTM, candidate DAG generator, DAG encoder, and memory network. The DAG generator extracted entities from questions, extended them to related entities, and gathered the entities as candidate answer entities. Also, it generated candidate DAGs, where subgraphs were generated whose relations and paths were related to questions. A key-value memory network with an attention layer was adopted to match between distributional representations of questions and candidate KB substructures. Experiments indicated that the proposed model had more advantages on complex question analysis. Focusing on answer single-relation questions over knowledge base, Qu et al. [66] proposed an attentive recurrent neural network with similarity matrix based convolutional neural network (AR-SNCNN), which further extended the encoder-compare framework by removing the entity matching model while preserving more information between words. In entity detection, question words were labeled by a Bi-LSTM and then extended to entity mentions. Candidate relations consisted of all the relations connected to the entities previously obtained, where relation detection included semantic level and literal level. In semantic level, the relations in Freebase were split into two parts and encoded separately, one indicated the type of subjects, while the other described relationship between subjects and objects. In literal level, in order to capture the similarity between same meaning in different expressions, a similarity matrix was constructed. Experiments showed that, with proposed extension method, the number of candidates was largely reduced but led to drop in recall.

A novel QA system called AQQUCN, was proposed in [71], which utilized knowledge from knowledge graph and corpus evidence to unify structured interpretation of questions with the ranking of response entities. AQQUCN has extended AQQU [5] by adding query corpus network (QCN), query-type network (QTN), query-relation network (QRN), and score combination network. QCN was used to assign a relevance score to each snippet, and QTN to output a compatibility score between the question and a candidate type. QRN outputed a compatibility score between a candidate relation and the question. In the score combination network, the scores from QCN, QTN, and QRN were combined.

Multi-hop reasoning in KBQA is still a challenging issue, requiring models to process through lengthy relations on knowledge base considering the given question, when in reality the exact number of hops to perform is unknown. With motivation to relax the number of hops restriction of models and reduce search space, Chen et al. [14] proposed an unrestricted-hop framework named UHop, which was compatible with models proposed in related work, hence allowed the installation of state-of-the-art models in UHop framework. Unrestricted-hop relation extraction in UHop was decomposed into two main subtasks: single-hop relation extraction and comparative termination decision. UHop worked in an iteration style: extract relations, transit to next entity, and decide whether to halt. While relation extraction process was modeled as classification, termination decision was treated as a comparison for being conducted using the same model. Additionally, in the UHop framework, dynamic question representation was adopted, regarding that the information already considered by previous selected relations. Two subtasks were jointly trained. Experiment results showed that the performance of models within UHop framework was comparable to those independent of it, with the unrestricted number of relation hops and reduced search space. With similar motivation, Lan et al. [48] introduced a method to incrementally construct relation paths leading to answer entities. Iterative path growth was based on beam search, iteratively adding relations into candidates set following a given algorithm. The termination was determined by comparing a calculated score with a threshold. For similarity calculation between questions and candidate relation paths, this work designed an iterative sequence matching model instead of changing representation. In each iteration, the last added relation was solely considered and all entities along the path were ignored. In this manner, the matching model computed scores without revisiting the earlier relations in a path. A scalar value was introduced into the framework to keep previous matching information.

To tackle the complexity of questions with constraints and with multiple hops of relations at the same time, Lan and Jiang [46] modified the staged query graph generation method by incorporating constraints when extending relation paths to further reduce search space. The generation method was based on beam search following defined actions: extend, connect, and aggregate action, where the extend action could be applied after the connect and aggregate actions. Query graph ranking was produced by a fully connected layer, into which 7-dimensional feature of candidate query graph was fed. Reinforce algorithm was used in ranking model training, while BERT-based semantic matching model was utilized in deriving features.

We conclude the DL structures, datasets, and evaluation metrics adopted in the methods for question answering over knowledge bases as Table 4.

Table 4 The DL structures, datasets, and evaluation metrics adopted in the methods for question answering over knowledge bases

Work	Network structures	Datasets	Metrics
Hao et al. [31]	RNNs and memory networks	Freebase, SPADES	F1
Luo et al. [58]	Bi-GRU	ComplexQuestions, WebQuestions and SimpleQuestions	F1
AR-SNCNN [66]	Bi-LSTM, Bi-GRU and CNN	SimpleQuestion	Accuracy, Recall
Lan et al. [48]	Bi-LSTM	MetaQA, PathQuestion and WC2014	%hits@1
Lan and Jiang [46]	BERT	ComplexWebQuestions, WebQuestionsSP and ComplexQuestions	Prec@1, F1
Jiang et al. [38]	LSTM-CRF and a multi-feature logistic model	GeoKG	-
Zhu et al. [114]	A tree-based LSTM	WebQuestions and WebQuestionsSP	Recall, F1
Lan et al. [47]	Matching-aggregation framework	WebQuestions, SimpleQuestions	F1, Accuracy
AQQUCN [71]	Convolutional networks	ClueWeb09B, TREC-INEX-KW, TREC-INEX, WebQuestions-KW and WebQuestions	F1, MAP, MMR and NDCG@10

6 Question generation

Question generation (QG), aiming at generate factoid question automatically, is receiving increasing interesting in recent years [63]. The researches in QG have been boosted since it can benefit model training [21, 37] and various QA tasks such as dialog systems. The underlying knowledge source for QG could be raw text or knowledge bases. Typical paradigm extracts keywords first, and then extends them into natural language sentences. Adopting end-to-end architectures, deep learning techniques enable these tasks performed jointly. The most used sequence-to-sequence frameworks follow an encoder–decoder structure, which often involves attention and copying mechanism [30]. We introduce those methods generate factoid questions over knowledge graphs first, followed by the methods based on unstructural texts.

Indurthi et al. [37] used a RNN-based model with LSTM units to generate question-answer pairs from knowledge graph. A set of keywords and answers were extracted from knowledge graph following rule-based methods. Hence, the generation task could be regarded as a sequence-to-sequence problem, from sequence consists of keywords to question in natural language, leading to adoption of an encode-decoder structure. Although the model was intended to be applied on generation over knowledge graph, cQA dataset was utilized instead of using knowledge graph in training phase.

Zero-shot question generation over knowledge graphs aims to generate questions involving predicates, subject types or object types unseen in training phase. For generating single-hop questions, Elsahar et al. [25] presented a neural model based on knowledge base triples and textual contexts. Motivated by the fact that manually written questions would follow existed sentences containing the same entities and the predicates occurred in the fact, this work paired input KB triples with a set of textual contexts. The contexts included a phrase containing a predicate, derived from Wikipedia, and two phrases containing entity types. The generative model adopted encoder–decoder architecture. While fact encoder with attention mechanism performed an attention-based encoding of KB triplet using TransE, textual context encoder used multiple GRUs to encode each textual context separately. Decoder utilized another GRU with an attention mechanism over all the textual contexts implicit representations. To tackle OOV problem, this work extended the copy actions with part-of-speech tags, utilizing the part-of-speech information to align the input and output texts. The reported highest score in experiments on SimpleQuestion was +2.39 BLEU-4 than the encoder–decoder baseline. Experiments showed that the proposed methods were effective in dealing with unseen predicates, but lost naturalness in some cases, which still needed further modified.

Kumar et al. [45] constructed a transformer-based model with self-attentive to automatically generate difficulty-controllable multi-hop questions. Under an encoder–decoder structure, the embedding of subgraph, answer encoding, and difficulty estimation were fed into encoder, while decoder took question, positional encoding, and difficulty encoding as input. The estimation of difficulty was based on the confidence of named entity recognition and linking performed and selectivity of questions. This

method generated complex questions conditioned on difficulty level in an end-to-end style.

Subramanian et al. [78] introduced a two-stage neural model for automatic question generation from documents. The model involved a key phrase extraction component in extracting key phrases from the document and a question generation component in constructing the question based on a key phrase. In the key phrase extraction component, given the fact that not all entities could lead to intended questions, they proposed a neural entity selection model to select entities. They also proposed another neural model, which utilized a pointer network to point the start and end boundaries of key phrases, to extract human-selected answer phrases from entity-less documents. In the question generation component, they adopted an attention-based translation model, a sequence-to-sequence framework with attention mechanism, and pointer-softmax mechanism to copy a word from the document and generate the word from a vocabulary.

However, existing neural question generation models, which utilize answer position feature to incorporate the answer information, have a serious problem that many generated questions contain same words in the answer, because sequence-to-sequence model tends to include all information from passages. Therefore, to prevent this tendency and generate intended questions, Kim et al. [42] proposed a novel model named answer-separated seq2seq which could utilize information from answer and passage by separating answer from the original passage. Different from RNN encoder–decoder model, the proposed model applied two one-layer Bi-LSTMs as encoders to separately extract contextual features from passages and answers. An answer-separated decoder was designed based on LSTM, involving a new module named keyword-net to extract keyword features of answers. To be more specific, the decoder used contextual features of passages from an attention mechanism and key information from answers extracted by keyword-net in every decoding step. In this way, the decoder utilized both the information from passages and answers to generate a question. Furthermore, a retrieval style word generator was applied as the decoder output layer in order to better capture word semantics.

Much previous research for question generation relies on sequence-to-sequence model and takes passages as input. As a result, target answers and hard-core answer position were neglected. To tackle this problem, Song et al. [76] developed a novel model based on sequence-to-sequence model with copy mechanism. They first applied two Bi-LSTMs to separately encode passages and answers, and then used a multi-perspective context matching algorithm on top of Bi-LSTM outputs to verify whether all passage words belong to the context of answers. Specifically, three match strategies were applied including full-matching, attentive-matching, and max-attentive-matching which correspondingly considered the last hidden state of answers, all words in answers without word order, and answer states that most relevant to passage states. Compared with baseline models, the model enabled matching states to contain matching information of all passage words and answer positions. Sufficient matching information enabled the decoder to generate more relevant questions. The matching information was added to the attention memory of the decoder. With sufficient matching information, the decoder was enabled to generate more relevant questions.

Intuitively, incorporating a whole paragraph as context should be more helpful to generate questions than solely using several sentences. However, irrelevant information is also brought in, leading to drops in performance [103]. To address this challenge, Zhao et al. [111] introduced a maxout pointer mechanism with gated self-attention networks. Following RNN-based encoder–decoder structure with LSTM units, gated self-attention was devised to aggregate information and capture intra-passage dependency after encoding steps, whose effectiveness in promoting important information was validated in experiments. Answer tagging was performed to indicate whether each word is in answer or not. An attention mechanism and a maxout pointer mechanism were applied in decode steps. Maxout pointer, as a modified copy/-pointer mechanism, intended to alleviate the repetition problem by setting limitation to repeated words.

We compare the performance of different methods for QG on unstructural texts and the results are shown in Table 5.

7 Datasets and evaluation

7.1 Datasets

We list seven recently used datasets for document-based QA tasks and eight for knowledge base QA tasks, as shown in Tables 6 and 7. With the relative new datasets, the effectiveness of different approaches for existed QA tasks can be better validated, while potential approaches for new QA tasks can be studied and explored.

7.1.1 Datasets for document-based QA

cMedQA v2.0 [107] is a dataset for Chinese Medical Question Answering. Collected from an online Chinese Medical Question Answering forum, the dataset consists of training set, development set and test set, summing up to 108,000 questions and 203,569 answers in total. Each question may have multiple different answers.

Table 5 Performance of QG on unstructural texts

Datasets	Methods	BLEU-4	$ROUGE_L$	METEOR
SQuAD Split-1 [22]	ASs2s [42]	16.20	43.96	19.92
	M2S+cp[76]	13.98	42.72	18.77
	Zhao et al. [111]	16.38	44.48	20.25
SQuAD Split-2 [113]	Subramanian et al. [78]	10.4	-	-
	ASs2s [42]	16.17	-	-
	M2S+cp[76]	13.91	-	-
	Zhao et al. [111]	16.85	44.99	20.62

Table 6 The list of document-based QA datasets

Dataset /Contributor		Domain	Task	Answer Type	Additional Features	Statistics	
						#questions	#answers
cMedQA v2.0 [107]		Medical	cQA	Human generated	No	108K	203,569
WebMedQA [32]		Medical	cQA	Human generated	Category	63,284	316,420
MedQuAD [6]		Medical	cQA	Human generated	Topic, synonyms, question type	47,457	47,457
StackExchange cQA Dataset [35]		Open	cQA	Human generated	212-dimensions	20,278	82,260
MCTest [35]	MC160	Open	Reading Comprehension	Multiple choices	No	640	2560
	MC500					2K	8K
BioMedical Knowledge [41]	BMKC_T	Biomedical	Machine Comprehension	Named Entities	No	473,127	-
	BMKC_LS					369,780	-
TOEFL Listening Comprehension Test [52]		Open	Reading Comprehension	Multiple choices	No	963	3852

Table 7 The list of knowledge base QA datasets

Dataset / Contributor	Query	Target KG	Size	Paraphrase	Multi-hop Questions	Statistics	
						#entities	#relations
LC-QuAD 2.0 [23]	SPARQL	Wikidata, DBpedia2018	30K	Yes	Included	21,258	1310
WebQuestions [8]	-	Freebase	5,810	Yes	No	–	–
WebQuestionsSP [101]	SPARQL	Freebase	4,737	No	No	–	–
MetaQA [110]	–	WikiMovies	400K	Yes	Included	43,233	-–
WorldCup2014 [105]	–	FIFA World Cup 2014	10K	No	Included	1,127	6
PathQuestion [112]	–	Freebase	7106	Yes	Included	2215	14
PathQuestion-Large [112]			2625			5035	364
SPADES [10]	–	Freebase	93K	No	No	4K	–
ComplexWebQuestions v1.1 [81]	SPARQL	Freebase	34K	No	Included	–	–

WebMedQA [32] is another Chinese Medical Question Answering dataset, containing 63,284 questions and 316,420 answers in total. Except from one best-adopted answer, each question also includes 4 negative answers for researches such as answer ranking and recommendation. MedQuAD [6] includes 47,457 medical question–answer pairs in English with additional annotations, such as the question type, the question focus, and so on.

Developed for biomedical machine comprehension tasks, BioMedical Knowledge Comprehension [41]

contains BMKC_T and BMKC_LS. With the abstract of biomedical domain academic papers being the context, the questions in BMKC_T and BMKC_LS are constructed from the title and the last sentence in the abstract of papers respectively. The number of queries in BMKC_T and BMKC_LS is 473,127 and 369,780 in total, respectively.

MCTest [35] is a reading comprehension dataset, consisted of MC160 and MC500 containing 160 and 500 texts respectively. Texts in dataset are fictional children's stories and each text has four multiple-choice questions, each with four answer candidates. TOEFL Listening Comprehension Test [52] includes 963 examples in total, where each example consists of a story, a question, and four choices. Each story includes audio recording and manual transcriptions.

StackExchange cQA Dataset [35] is a cQA dataset for answer selection, including 20,278 questions and their corresponding 82,260 answers with 212-dimensional features extracted.

7.1.2 Datasets for knowledge base QA

LC-QuAD 2.0 (Large-Scale Complex Question Answering Dataset) [23] contains 30K questions with paraphrases and corresponding SPARQL queries over Wikidata and DBpedia2018. Questions in LC-QuAD 2.0 cover 10 different types providing variety and complexity.

WebQuestions [8] is a popular benchmark dataset, containing 5,810 questions from web, 84% of which can be answered with a single relation. WebQuestionsSP [101] re-annotates the questions in WebQuestions, providing both semantic parses and derived answers, while the questions with incomplete parses were removed during annotation.

ComplexWebQuestions, released by Talmor and Berant [81], contains 34,689 broad and complex questions with corresponding SPAQRL queries based on WebQuestionsSP for reducing the leakage from training to test data [82].

MetaQA (MoviE Text Audio QA) [110] is a dataset for movie domain, containing a mixture of 1-hop to 3-hop questions. WorldCup2014 [105] contains 10K questions over a built knowledge base of football players, requiring either 1-hop or 2-hop reasoning.

PathQuestion [112] and PathQuestion-Large [112] are based on two subsets of Freebase, with 2-hop path or 3-hop path. SPADES (Semantic PArsing of DEclarative Sentences) [10] contains 93K fill-in-the-blank cloze-styled questions and 1.8M entities collected from web. Empty slots in the sentences are obtained by removing an entity randomly.

7.2 Evaluation metrics

There have been various evaluation metrics for QA tasks. Here in table 8, the most widely used evaluation measures are introduced, including precision, recall, F1, accuracy, ER, EM, MAP, and MRR.

8 Discussion

Question answering has been widely applied in different areas. We have noticed that there have been many works focus on QA in medical domain, offering answers based on given question–answer pairs set. In this way, the information in the online medical community can be fully utilized. Other closed-domain question answering systems also acquire promising performance, making the question answering style query possible. Meantime, some works explore the potential of models to retrieve answers in a given document, which may serve as a more efficient search engine. QA techniques provide a more friendly way for human–machine interaction, which can be further developed into the digital assistant or other applications.

8.1 Typical deep learning methods in QA

At the beginning of the deep learning (DL) applications in QA, the common approach was to adopt a basic type of deep learning architecture and train it with downstream tasks. Over the past few years, the frameworks have become more complicated, combining methods for complex tasks. These frameworks usually involved several different basic DL architectures and incorporated the non-DL methods. These combinations made the frameworks more sophisticated and perform better while lost generalization relatively. However, it is still unclear how architectures for basic tasks can be combined to perform well in applied tasks [62].

In this section, we briefly summarize the main deep learning methods adopted in different question answering tasks.

8.1.1 RNNs-based methods

Recurrent neural network (RNN) [24] is a kind of neural network where the input of each unit not only takes the network input but also considers the output of previous units. Due to its structure nature, multilayer RNNs are able to deal with text and capture hierarchical patterns. Bidirectional RNNs (Bi-RNNs) combine two RNNs in opposite direction (forward and backward) to fully utilize past and future information for each time frame. Bi-RNNs take

Table 8 The widely used evaluation metrics for QA tasks

Metrics	Description	Formulas
Precision	TP, FP, TN, FN stand for true positive, false positive, true negative, and false negative, respectively.	$Precision = \frac{TP}{TP+FP}$
Recall		$Recall = \frac{TP}{TP+FN}$
F1		$F1\text{-}socre = 2 \times \frac{Precision \times Recall}{Precision+Recall}$
Accuracy	N is the total number of examples.	$Accuracy = \frac{TP+TN}{N}$
Error rate		$ErrorRate = \frac{FP+FN}{N}$
Exact match (EM)	EM measures the percentage of predictions which exactly match the answers.	-
Mean average precision (MAP)	MAP is used to evaluate the systems that return a ranked sequence of documents, questions, or answers. For a set of queries, it represents the mean of the Average Precision (AvgP) values for each query.	$MAP = \frac{1}{N}\sum_{i=1}^{N} AvgP_i$
Average precision (AvgP)	AvgP is the Average Precision for a query in query set. n represents the total number of relevant documents. $P(k)$ is the precision of the top k ranked list, and $r(k)$ is an indicator function being assigned 1 if item k is relevant.	$AvgP = \frac{\sum_{k=1}^{n}(P(k)\times r(k))}{n}$
Mean reciprocal rank (MRR)	MRR is often used as a measurement in multi-results problems. Q indicates the total number of answers and $rank_i$ is the location of the ground-truth answer in ranking sequence.	$MRR = \frac{1}{\|Q\|}\sum_{i=1}^{Q}\frac{1}{rank_i}$

whole sentences into account rather than only considering former information. Long short-term memory model (LSTM) [34], an improved version of RNNs, introduces the memory mechanism and enhances the ability of the network to process semantic relations in long text sequence. LSTM has shown its effectiveness in various tasks, and thus is widely adopted in deep learning research areas. As a simplified variant of LSTM, RNN with gated recurrent unit (GRU) [15] is with fewer parameters and more efficient in training. There have been works on exploring the effectiveness of adopting RNNs-based methods for QA systems and further improving these methods.

8.1.2 CNNs-based methods

Convolutional neural network (CNN) [26, 50] is effective in capturing local pattern [28]. The output of higher layer in CNNs can represent more abstract features of longer sequences in inputs [85]. Also, CNNs can detect local patterns without taking their positions into account. Therefore, CNNs have been widely taken for various tasks in QA.

8.1.3 Attention mechanism

Attention mechanism [3], motivated by human visual cognition nature, enables the models to pay more attention to important features and leave out trivial information. It can be interpreted as performing a weighted sum of input vectors, where the weights are learned automatically [60]. Attention mechanisms have been proved effective on various tasks and become popular among different network architectures.

8.1.4 Hybrid methods

A hybrid method usually consists of several different methods. The hybrid method can be a combination of deep learning methods or a combination of deep learning methods with non-deep learning methods. These methods are usually more complex, requiring more domain knowledge to build. Therefore, they can be more effective in specific tasks. There have been works on taking the advantages of DL methods together by combining them. However, since deep learning involves multilayer network, which contains a large number of parameters, training phase usually requires a considerable amount of labeled data. Therefore, many studies tried to address this issue by incorporating non-DL methods into the framework. On the other hand, for some specific tasks, DL methods not always perform the best but still have some merits. The combinations may take the strengths together. We summarize the methods adopted in the hybrid frameworks in Table 9.

8.1.5 Methods statistics

This paper has presented a statistical analysis on the deep learning methods used in the literature, as shown in Figure 1. The RNNs architecture is the most popular deep network structure, while the attention mechanism is widely adopted in different methods.

Table 9 The DL structures adopted in hybrid methods

Work	Methods	Task
Zhang et al. [107]	Bi-GRUs and CNNs with multi-attentive interaction	Chinese medical question answer selection
Khalifa and Shaalan [40]	CNNs and LSTM-CRF	Arabic NER task
Yang and Fan [98]	CNNs and Memory Networks	The multi-hop text-based QA problem
Shao et al. [74]	CNNs and LSTM	Answer selection
Sun and Xia [80]	CNNs and LSTM	Question answering of specific low resources languages
Hao et al. [31]	RNNs and Memory Networks	Knowledge based QA
Jiang et al. [38]	LSTM-CRF and multi-feature logistic model	The semantic querying of geographic knowledge
Li et al. [54]	Logic Programming and attention-based Bi-LSTM	Comparative questions answering
Croce et al. [17]	The kernel methods and neural networks	Question classification

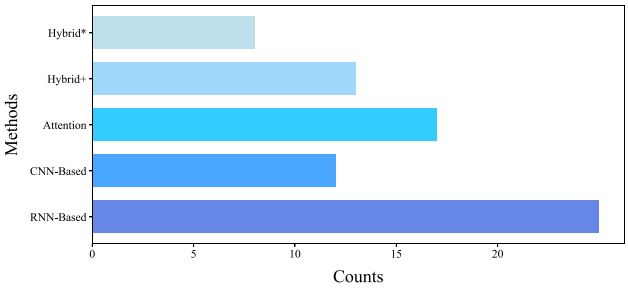

Fig. 1 Data statistics of deep learning methods. Hybrid^{+} stands for the methods combining different deep learning architectures. Hybrid* stands for the methods incorporating both deep learning architectures and non-DL methods

8.2 Performance comparison

We compare the performance of different methods on several popular QA datasets and the results are shown in Table 10. On the Wiki-QA dataset, SBiLSTM-coA achieves the highest MRR performance as 0.8401 but has slight lower MAP as 0.7613 compared with wGRU-sGRU-G_{l2}-Cn. QA-TK$^{*}_{AP}$ has the lowest performance on both MAP and MRR. On the TREC-QA dataset, MSDCNN-5, depCNN, and SBiLSTM-coA have relative comparable performance with MAP between 0.76–0.78 and MRR between 0.77–0.85. Although some methods do not yield state-of-the-art performance, the innovations of the methods particular on network modification and combination are still promising.

Table 10 Performance of deep learning-based QA methods

Datasets	Methods	MAP	MRR	Error Rates	Acc	EM	F1
Wiki-QA[1]	SBiLSTM-coA [11]	0.7613	0.8401	–	–	–	–
	wGRU-sGRU-G_{l2}-Cn [83]	0.7638	0.7825	–	–	–	–
	QA-TK$^{*}_{AP}$ [73]	0.6941	0.7077	–	–	–	–
	P-CNN [77]	0.7347	0.7371	–	–	–	–
bAbI [91]	GORU [39]	-	-	-	60.4	–	–
bAbI-1k	*t*-MEM-NN [86]	–	–	–	–	–	–
	CMemN2N [98]	–	–	–	0.8706	–	–
bAbI-10k	EnDMN [104]	–	–	minimum:0.3	–	–	–
	CMemN2N [98]	–	–	–	0.9505	–	–
	MemN2N-GL [93]	–	–	mean:0.19	–	–	–
SQuAD [67]	SFM [75]	-	-	-	-	79.0	86.2
	wGRU-sGRU-G_{l2} -Cnt [83]	0.9209	0.9209	-	–	–	–
	APIP$_5(n_i = 5)$ [64]	–	–	–	–	78.53	86.40
TREC-QA[87]	MSDCNN-5 [56]	0.762	0.824	-	–	-	–
	depCNN [108]	0.7669	0.8215	–	–	–	–
	SBiLSTM-coA [11]	0.7643	0.7751	–	–	–	–

[1] https://www.microsoft.com/en-us/download/details.aspx?id52419

It is noticeable that performances vary on different datasets due to different tasks requirements. For example, although Wiki-QA dataset and TREC-QA dataset contained factoid questions are both for reading comprehension task, the average performance on Wiki-QA is lower than on TREC-QA. While TREC-QA expects single sentences as answers, Wiki-QA requires inference across several sentences when necessary. Also, TREC-QA contains more questions which would be helpful to train deep learning models. Hence, the performances gaps among different datasets may derive from the complexity of the collected data, scale of dataset, task requirements, and so on. As a result, it is supposed that a newly proposed method should be validated on different datasets.

8.3 Challenges and opportunities

Deep learning techniques have boosted the performance of question answering tasks over the last few years. Many different network architectures have been applied, bringing the large progress. However, there still remain many challenges. These challenges could, in turn, be the opportunities for the development of DL-based QA in the future.

*Pretrained Language Models.*The pretrained language models have made big progress in recent years, such as BERT [19], ELMo [65], and XLNet [99]. Pretrained language models can be obtained by unsupervised training, which is able to fully utilize massive available unstructured plain text. By incorporating pretrained models, the performance of QA tasks may be further improved [27, 46].

Semantics Matter More. There have been many attempts to utilize more semantics. Many works have adopted attention mechanisms to enable the models to focus on keywords [9], which are relevant to semantics of sentences. Some have worked on capturing more semantics in obtained sentence representations [35].

Framework Design. A number of newly proposed frameworks integrate different deep learning methods. However, how to maximize the performance of the framework is still under study. Rather than simply combining different network structures, some researchers dived into network architectures design. Different gate mechanisms in RNNs lead to distinctions in memory ability [39, 93], while different designs of the convolution operation in CNNs present different performance [56, 77]. Designing network architectures is more tricky, which requires researchers to have a deeper insight into deep learning frameworks.

QA on Small-Scale Dataset. Although there have been a lot of datasets collected and released for QA tasks, the QA researches of low-resource languages suffer from the lacking of large-scale datasets. On one hand, there remains a need for new datasets on minority languages for QA tasks [80] . On the other hand, novel techniques are desired to build decent models from a small-scale dataset. It may involve few-shot learning, zero-shot learning techniques or transfer learning techniques [25, 37, 94].

Low Computation Cost. Building a deep learning model requires a long training phrase and consumes a lot of computation resources. If the model architecture is elaborately designed with fewer parameters to update, it may save plenty of training time and resources. This can be achieved by exploring simplified architectures or using model compression techniques. Efforts have been made to increase training efficiency, such as [49, 70], and [84]. Some have tried to combine machine learning techniques with deep neural networks to obtain a more efficient network [17].

QA with External Knowledge. Various tasks in NLP have made attempts to promote the performance by incorporating external knowledge, such as [92, 96] and [95]. However, many studies on QA did not fully utilize external knowledge bases, but merely rely on plain text or datasets. There have been efforts to introduce external knowledge into QA. Yang et al. [97] leveraged the information from knowledge base for cQA by introducing knowledge graph embedding into representation learning. With similar motivation, Huang et al. [36] utilized external knowledge in answer selection task. In addition to explicit use of external knowledge, incorporating pretrained models can also enable the models to better capture semantic similarity.

9 Conclusion

This paper reviews recently proposed deep learning methods for question answering. The survey covers the recent literature, datasets, evaluation metrics, typical deep learning models, performance comparison, as well as challenges and opportunities. First, we review and discuss studies leveraging deep learning methods on QA tasks including question classification, answer extraction, question–answer matching, knowledge base question answering, and question generation. We then list recently used datasets for QA tasks with descriptions and statistics, and introduce widely used evaluation metrics. After that, we provide a brief introduction to deep learning methods. Finally, we present performance comparison, current challenges, and future opportunities in deep learning-based question answering.

Acknowledgements This work is supported by grants from National Natural Science Foundation of China (No. 61772146), The Science and Technology Plan of Guangzhou (No. 201804010296), and Natural Science Foundation of Guangdong Province, China (No. 2018A030310051).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Adlouni YE, Rodríguez H, Meknassi M, El Alaoui SO, Ennahnahi N (2019) A multi-approach to community question answering. Expert Sys Appl 137:432–442
2. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Hasan M, van Essen BC, Awwal AAS, Asari VK (2019) A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3):292
3. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: ICLR
4. Banerjee S, Naskar S, Rosso P, Bandyopadhyay S (2018) Code mixed cross script factoid question classification - a deep learning approach. J Intell & Fuzzy Sys 34(5):2959–2969
5. Bast H, Haussmann E (2015) More accurate question answering on freebase. In: CIKM'15, pp 1431–1440
6. Ben Abacha A, Demner-Fushman D (2019) A question-entailment approach to question answering. BMC Bioinfo 20(1):e33
7. Bengio Y (2009) Learning deep architectures for AI. Found Trends in Machine Learn 2(1):1–127
8. Berant J, Chou A, Roy F, Liang P (2013) Semantic parsing on freebase from question-answer pairs. In: EMNLP, pp 1533–1544
9. Bi M, Zhang Q, Zuo M, Xu Z, Jin Q (2019) Bi-directional lstm model with symptoms-frequency position attention for question answering system in medical domain. Neural Process Lett 51(5):570
10. Bisk Y, Reddy S, Blitzer J, Hockenmaier J, Steedman M (2016) Evaluating induced ccg parsers on grounded semantic parsing. In: EMNLP, pp 2022–2027
11. Cai L, Zhou S, Yan X (2019) Yuan R (2019) A stacked bilstm neural network based on coattention mechanism for question answering. Computat Intell Neurosci 9:1–12
12. Cai LQ, Wei M, Zhou ST, Yan X (2020) Intelligent question answering in restricted domains using deep learning and question pair matching. IEEE Access 8:32922–32934
13. Chen Z, Zhang C, Zhao Z, Yao C, Cai D (2018) Question retrieval for community-based question answering via heterogeneous social influential network. Neurocomputing 285:117–124
14. Chen ZY, Chang CH, Chen YP, Nayak J, Ku LW (2019) Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In: NAACL-HLT, pp 345–356
15. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP, pp 1724–1734
16. Cortes E, Woloszyn V, Binder A, Himmelsbach T, Barone D, Möller S (2020) An empirical comparison of question classification methods for question answering systems. In: LREC, pp 5408–5416
17. Croce D, Filice S, Basili R (2019) Making sense of kernel spaces in neural learning. Computer Speech & Language 58:51–75
18. Dargan S, Kumar M, Ayyagari MR, Kumar G (2019) A survey of deep learning and its applications: A new paradigm to machine learning. Archi Computat Method Eng 85(4):114
19. Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp 4171–4186
20. Dimitrakis E, Sgontzos K, Tzitzikas Y (2019) A survey on question answering systems over linked data and documents. J Intell Info Sys 51(5):570
21. Dong L, Mallinson J, Reddy S, Lapata M (2017) Learning to paraphrase for question answering. In: EMNLP, pp 875–886
22. Du X, Shao J, Cardie C (2017) Learning to ask: Neural question generation for reading comprehension. In: ACL, pp 1342–1352
23. Dubey M, Banerjee D, Abdelkawi A, Lehmann J (2019) Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. SEMWEB 11779:69–78
24. Elman JL (1990) Finding structure in time. Cognitive Sci 14(2):179–211
25. Elsahar H, Gravier C, Laforest F (2018) Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In: NAACL-HLT, pp 218–228
26. Fukushima K (1988) Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Networks 1(2):119–130
27. Garg S, Vu T, Moschitti A (2020) Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. AAAI 34:7780–7788
28. Goldberg Y (2016) A primer on neural network models for natural language processing. J Artif Intell Res 57(1):345–420
29. Green BF, Wolf AK, Chomsky C, Laughery K (1961) Baseball: an automatic question-answerer. In: IRE-AIEE-ACM '61 (Western), pp 219–224
30. Gulcehre C, Ahn S, Nallapati R, Zhou B, Bengio Y (2016) Pointing the unknown words. In: ACL, pp 140–149
31. Hao Z, Wu B, Wen W, Cai R (2019) A subgraph-representation-based method for answering complex questions over knowledge bases. Neural Networks 119:57–65
32. He J, Fu M, Tu M (2019) Applying deep matching networks to chinese medical question answering: a study and a dataset. BMC Med Info Decision Making 19(S2):1
33. Hirschman L, Gaizauskas R (2001) Natural language question answering: the view from here. Nat Lang Eng 7(4):275–300
34. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computat 9(8):1735–1780
35. Huang H, Wei X, Nie L, Mao X, Xu XS (2019) From question to text: Question-oriented feature attention for answer selection. ACM Trans Info Sys 37(1):1–33
36. Huang W, Qu Q, Yang M (2020) Interactive knowledge-enhanced attention network for answer selection. Neural Comput Appl 32(15):11343–11359
37. Indurthi SR, Raghu D, Khapra MM, Joshi S (2017) Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model. In: EACL, pp 376–385
38. Jiang B, Tan L, Ren Y, Li F (2019) Intelligent interaction with virtual geographical environments based on geographic knowledge graph. ISPRS Int J Geo-Info 8(10):428
39. Jing L, Gulcehre C, Peurifoy J, Shen Y, Tegmark M, Soljacic M, Bengio Y (2019) Gated orthogonal recurrent units: on learning to forget. Neural Computat 31(4):765–783
40. Khalifa M, Shaalan K (2019) Character convolutions for arabic named entity recognition with long short-term memory networks. Comp Speech & Language 58:335–346
41. Kim S, Park D, Choi Y, Lee K, Kim B, Jeon M, Kim J, Tan AC, Kang J (2018) A pilot study of biomedical text comprehension using an attention-based deep neural reader: design and experimental analysis. JMIR Med Info 6(1):e2
42. Kim Y, Lee H, Shin J, Jung K (2019) Improving neural question generation using answer separation. AAAI 33:6602–6609
43. Kolomiyets O, Moens MF (2011) A survey on question answering technology from an information retrieval perspective. Info Sci 181(24):5412–5434

44. Kumar A, Irsoy O, Ondruska P, Iyyer M, Bradbury J, Gulrajani I, Zhong V, Paulus R, Socher R (2016) Ask me anything: Dynamic memory networks for natural language processing. In: ICML, pp 1378–1387
45. Kumar V, Hua Y, Ramakrishnan G, Qi G, Gao L, Li YF (2019) Difficulty-controllable multi-hop question generation from knowledge graphs. SEMWEB 11778:382–398
46. Lan Y, Jiang J (2020) Query graph generation for answering multi-hop complex questions from knowledge bases. In: ACL, pp 969–974
47. Lan Y, Wang S, Jiang J (2019) Knowledge base question answering with a matching-aggregation model and question-specific contextual relations. IEEE/ACM Trans Audio, Speech, and Language Process 27(10):1629–1638
48. Lan Y, Wang S, Jiang J (2019) Multi-hop knowledge base question answering with an iterative sequence matching model. In: ICDM, pp 359–368
49. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2020) Albert: A lite bert for self-supervised learning of language representations. In: ICLR
50. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceed of the IEEE 86:2278–2324
51. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
52. Lee CH, Lee HY, Wu SL, Liu CL, Fang W, Hsu JY, Tseng BH (2019) Machine comprehension of spoken content: Toefl listening test and spoken squad. IEEE/ACM Trans on Audio, Speech, and Language Process 27(9):1469–1480
53. Li J, Sun A, Han J, Li C (2022) A survey on deep learning for named entity recognition. IEEE Transact Knowledge & Data Eng 34:50–70
54. Li X, Zhang S, Wang B, Gao Z, Fang L, Xu H (2019) A hybrid framework for problem solving of comparative questions. IEEE Access 7:185961–185976
55. Lin T, Goyal P, Girshick R, He K, Dollár P (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Machine Intell 42(2):318–327
56. Liu D, Niu Z, Zhang C, Zhang J (2019) Multi-scale deformable cnn for answer selection. IEEE Access 7:164986–164995
57. Liu H, Liu Y, Wong LP, Lee LK, Hao T (2020) A hybrid neural network bert-cap based on pre-trained language model and capsule network for user intent classification. Complexity 2020:1–11
58. Luo K, Lin F, Luo X, Zhu K (2018) Knowledge base question answering via encoding of complex query graphs. In: EMNLP, pp 2185–2194
59. Mahmoud A, Zrigui M (2019) Sentence embedding and convolutional neural network for semantic textual similarity detection in arabic language. Arab J Sci Eng 44(11):9263–9274
60. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning-based text classification: A comprehensive review. ACM Comput Surv 54(3):62:1–62:40
61. Ojokoh B, Adebisi E (2019) A review of question answering systems. J Web Eng 17(8):717–758
62. Otter DW, Medina JR, Kalita JK (2021) A survey of the usages of deep learning in natural language processing. IEEE Trans Neural Network Learn Sys 32:604–624
63. Pan L, Lei W, Chua TS, Kan MY (2019) Recent advances in neural question generation. ArXiv abs/1905.08949
64. Parshakova T, Rameau F, Serdega A, Kweon IS, Kim DS (2019) Latent question interpretation through variational adaptation. IEEE/ACM Trans Audio, Speech and Language Process 27(11):1713–1724
65. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: NAACL-HLT, pp 2227–2237
66. Qu Y, Liu J, Kang L, Shi Q, Ye D (2018) Question answering over freebase via attentive rnn with similarity matrix based cnn. arXiv: abs/1804.03317
67. Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016) Squad: 100,000+ questions for machine comprehension of text. In: EMNLP, pp 2383–2392
68. Ren Q, Cheng X, Su S (2020) Multi-task learning with generative adversarial training for multi-passage machine reading comprehension. AAAI 34:8705–8712
69. Roy PK, Singh JP (2019) Predicting closed questions on community question answering sites using convolutional neural network. Neural Comput Appl 19(5):53
70. Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv: abs/1910.01108
71. Sawant U, Garg S, Chakrabarti S, Ramakrishnan G (2019) Neural architecture for question answering using a knowledge graph and web corpus. Info Retr J 22(3–4):324–349
72. Shah AA, Ravana SD, Hamid S, Ismail MA (2018) Accuracy evaluation of methods and techniques in web-based question answering systems: a survey. Knowl Info Sys 58(03):611–650
73. Shao T, Guo Y, Chen H, Hao Z (2019) Transformer-based neural network for answer selection in question answering. IEEE Access 7:26146–26156
74. Shao T, Kui X, Zhang P, Chen H (2019) Collaborative learning for answer selection in question answering. IEEE Access 7:7337–7347
75. Shuang K, Liu Y, Zhang W, Zhang Z (2018) Summarization filter: Consider more about the whole query in machine comprehension. IEEE Access 6:58702–58709
76. Song L, Wang Z, Hamza W, Zhang Y, Gildea D (2018) Leveraging context information for natural question generation. In: NAACL-HLT, New Orleans, Louisiana, pp 569–574
77. Song Y, Hu QV, He L (2019) P-cnn: Enhancing text matching with positional convolutional neural network. Knowledge-Based Sys 169:67–79
78. Subramanian S, Wang T, Yuan X, Zhang S, Trischler A, Bengio Y (2018) Neural models for key phrase extraction and question generation. In: QA@ACL, pp 78–88
79. Sukhbaatar S, Szlam A, Weston J, Fergus R (2015) End-to-end memory networks. In: NIPS, p 2440-2448
80. Sun Y, Xia T (2019) A hybrid network model for tibetan question answering. IEEE Access 7:52769–52777
81. Talmor A, Berant J (2018) Repartitioning of the complexwebquestions dataset. arXiv: abs/1807.09623
82. Talmor A, Berant J (2018) The web as a knowledge-base for answering complex questions. In: NAACL-HLT, pp 641–651
83. Tan C, Wei F, Zhou Q, Yang N, Du B, Lv W, Zhou M (2018) Context-aware answer sentence selection with hierarchical gated recurrent neural networks. IEEE/ACM Trans Audio, Speech and Language Process 26(3):540–549
84. Tay Y, Tuan LA, Hui SC (2018) Hyperbolic representation learning for fast and efficient neural question answering. In: WSDM, pp 583–591
85. Tixier AJP (2018) Notes on deep learning for nlp. arXiv: abs/1808.09772
86. Tolias K, Chatzis SP (2019) *t*-exponential memory networks for question-answering machines. IEEE Trans Neural Networks Learn Sys 30(8):2463–2477
87. Wang M, A Smith N, Mitamura T (2007) What is the jeopardy model? a quasi-synchronous grammar for qa. In: EMNLP-CoNLL, pp 22–32

88. Wang S, Zhou W, Jiang C (2020) A survey of word embeddings based on deep learning. Computing 102(3):717–740
89. Wang Z, Liu J, Xiao X, Lyu Y, Wu T (2018) Joint training of candidate extraction and answer selection for reading comprehension. In: ACL, pp 1715–1724
90. Wen J, Tu H, Cheng X, Xie R, Yin W (2019) Joint modeling of users, questions and answers for answer selection in cqa. Expert Sys Appl 118:563–572
91. Weston J, Bordes A, Chopra S, Rush AM, van Merriënboer B, Joulin A, Mikolov T (2016) Towards ai-complete question answering: A set of prerequisite toy tasks. In: ICLR (Poster)
92. Wu Y, Wu W, Li Z, Zhou M (2018) Knowledge enhanced hybrid neural network for text matching. In: AAAI, pp 5586–5593
93. Wulamu A, Sun Z, Xie Y, Xu C, Yang A (2019) An improved end-to-end memory network for qa tasks. Computers, Materials & Continua 60(3):1283–1295
94. Xia C, Zhang C, Yan X, Chang Y, Yu P (2018) Zero-shot user intent detection via capsule neural networks. In: EMNLP, pp 3090–3099
95. Xin J, Lin Y, Liu Z, Sun M (2018) Improving neural fine-grained entity typing with knowledge attention. In: AAAI, pp 5997–6004
96. Yang B, Mitchell T (2017) Leveraging knowledge bases in lstms for improving machine reading. In: ACL, pp 1436–1446
97. Yang M, Tu W, Qu Q, Zhou W, Liu Q, Zhu J (2019) Advanced community question answering by leveraging external knowledge and multi-task learning. Knowledge-Based Sys 171:106–119
98. Yang X, Fan P (2019) Convolutional end-to-end memory networks for multi-hop reasoning. IEEE Access 7:135268–135276
99. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. In: NeurIPS, pp 5754–5764
100. Yao X (2014) Feature-driven question answering with natural language alignment. John Hopkins University (PhD thesis)
101. Yih Wt, Richardson M, Meek C, Chang MW, Suh J (2016) The value of semantic parse labeling for knowledge base question answering. In: ACL, pp 201–206
102. Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. IEEE Comput Intell Mag 13(3):55–75
103. Yuan X, Wang T, Gulcehre C, Sordoni A, Bachman P, Zhang S, Subramanian S, Trischler A (2017) Machine comprehension by text-to-text neural question generation. In: Rep4NLP@ACL, pp 15–25
104. Yue C, Cao H, Xiong K, Cui A, Qin H, Li M (2017) Enhanced question understanding with dynamic memory networks for textual question answering. Expert Sys Appl 80:39–45
105. Zhang L, Winn J, Tomioka R (2016) Gaussian attention model and its application to knowledge base embedding and question answering. arXiv: abs/1611.02266
106. Zhang S, Zhang X, Wang H, Cheng J, Li P, Ding Z (2017) Chinese medical question answer matching using end-to-end character-level multi-scale cnns. Appl Sci 7(8):767
107. Zhang S, Zhang X, Wang H, Guo L, Liu S (2018) Multi-scale attentive interaction networks for chinese medical question answer selection. IEEE Access 6:74061–74071
108. Zhang S, Zhang W, Niu J (2019) Improving short-text representation in convolutional networks by dependency parsing. Knowledge and Information Systems 61(1):463–484
109. Zhang X, Lu W, Li F, Peng X, Zhang R (2019) Deep feature fusion model for sentence semantic matching. Comput, Mater & Continua 61(2):601–616
110. Zhang Y, Dai H, Kozareva Z, Smola AJ, Le Song (2018) Variational reasoning for question answering with knowledge graph. In: AAAI, pp 6069–6076
111. Zhao Y, Ni X, Ding Y, Ke Q (2018) Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In: EMNLP, pp 3901–3910
112. Zhou M, Huang M, Zhu X (2018) An interpretable reasoning network for multi-relation question answering. In: COLING, pp 2010–2022
113. Zhou Q, Yang N, Wei F, Tan C, Bao H, Zhou M (2017) Neural question generation from text: A preliminary study. NLPCC 10619:662–671
114. Zhu S, Cheng X, Su S (2020) Knowledge-based question answering by tree-to-sequence learning. Neurocomputing 372:64–72