Interactive Question Answering Literature Review

Gregory Furman
FRMGRE001@myuct.ac.za
UCT Computer Science Honours

Abstract

Question-answering systems are said to be situated at the intersection between natural language processing and information retrieval. A fundamental challenge in creating conversationally plausible and increasingly accurate QA systems is that traditional NLP methods fail to capture the underlying structural composition of language. That is, models fail to *comprehend* and *understand* required tasks. In this paper, we explore both traditional methods of QA and outline the current state-of-the-art. To this end, we investigate the prevalence of machine learning techniques, namely unsupervised learning, as a possible means of capturing latent structural and compositional factors of natural language, in the form of conversations and textual-utterences. We also consider interactive question answering as a viable solution to the core challenge of natural language understanding. We also examine reinforcement learning as a means of solving complex QA problems which show positive improvements over traditional methods. Finally, we examine the integration of RL and text-based environments for IQA. The investigation finds that the structure of IQA problems makes them well suited for solving using RL. Based on the current literature, we suggest that state-of-the-art NLP methods such as *neural semantic parsing* and *machine reading comprehension*, would see consequential improvements in performance, dialectic ability, and accuracy over non-interactive systems. Moreover, we propose text-based game simulators as a demonstrably proficient and viable channel for future QA research.

1 Introduction

Question answering (QA) systems have become a popular tool for the navigation and querying of large-scale knowledge bases (KB) [16]. However, an increasing demand for speed and precision of such systems has motivated the exploration of differing methodologies and techniques [24, 44, 48]. *Interactive question answering* (IQA) has emerged as a hybrid of QA and dialogue systems that seek to initiate a dialogue with a user in order to better answer a query [45].

This paper aims to review the state-of-the-art of both IQA and QA in order to better elucidate strengths, weaknesses, and applicability of current methodologies and practices. To this end, current literature has indicated neural semantic parser, machine reading comprehension, named entity recognisers, recurrent neural networks, and reinforcement learning agents to be highly pertinent and useful techniques for answering domain specific user queries to a KB [26, 55].

We will also investigate the viability of text-based games, as opposed to traditional training methods, as means of developing more generalisable and novel systems for natural language processing (NLP) based QA systems [2].

2 Question Answering

Natural Language processing can be defined as the epistemological union between Computer Science, Linguistics, and Artificial Intelligence. It seeks to use a variety of cross-disciplinary techniques allow for machines to achieve some level of *comprehension* of natural languages. QA is said to be situated at the intersection between Information Retrieval (IR) and NLP [24]. Such systems see requests for information retrieval being (at least partially) expressed as natural language statements or questions [44]. This marriage of using IR to narrow down information and NLP techniques to extract answers has proven itself to be a powerful union. QA systems are determined to be either *domain-specific* or *open-domain*.

Domain-specific question answering is concerned with questions that fall within a clearly defined and limited domain [66]. This allows for domain-specific knowledge or previously formalised ontologies to be integrated with the QA process. On the contrary, open-domain QA systems are designed to answer questions relating to, ideally, any topic [66]. Thus, the system relies on generalised ontologies and a wide domain of knowledge to draw upon. Such systems usually require vast amounts of readily available data from which to extract answers.

Current state-of-the-art automated QA systems are typically organised into a trimodal architecture that includes: a question-analysis module, a search engine, and an answer extraction module [48, 62]. The question-analysis module classifies questions into types, extracts the pertinent keywords, and determines a questions *answer type*. This module has also been used to reformulate a question into a semantically equivalent expression or set of similar questions [28]. The output of this model is then fed into the search module wherein a subset of a corpus deemed most likely to contain the answer to the original question is returned. The answer extraction module then uses the question's answer type along with this narrowed down subset in order to generate a ranked-list of possible answers to the original question [60].

It is said that most QA systems rely on factoid questions [39]. That is, questions whose answers can be expressed with simple short-text. Jurafsky & Martin [39] outline two major paradigms for factoid-based QA: IR based QA (open-domain) and knowledge-based QA (domain-specific). These

approaches are said to approach QA using simple NLP & IR and QA using NLP-based reasoning, respectively [28].

3 Information Retrieval

Open-domain question answering (otherwise referred to as IR-based QA) aims to find an answer a to a question q from a large corpus of text D [61]. This system is said to generally follow the *retrieve and read* model [39]. That is, a two-stage process that requires an IR engine to **retrieve** a subset of relevant documents or passages D_r from D that are **read** and processed by some *reading comprehension* system that finds spans (a continuous string of text) deemed most likely to answer the question [39].

3.1 Architecture

Typically, the retrieve stage follows classic IR techniques that aim to narrow down relevant passages or documents from a vast corpus of data. These retrieved passages are processed by NLP-based neural reading comprehension algorithms that aim to return a ranked-list of documents containing possible answers to the original question [60]. IR-based QA systems are assumed to have a large corpus of data to draw upon, a search module/engine for indexing, a named entity recogniser, a standard and extendable set of types useful to a wide range of domains, as well as *question-analysis* and *answer-extraction* modules [60]. The composition of these components can be seen in Figure 1.

3.1.1 Reading Comprehension. Span labeling is commonly used to model the job of answer-extraction. Thus a *neural algorithm for reading comprehension* (reader) is given a passage p and a question q and assigns probabilities to each span a such that it is an answer to q.

More formally, given a question q of n tokens $(q_1, ..., q_n)$ and a passage p of m tokens $(p_1, ..., p_m)$, the reader will be required to calculate $Pr(a|q, p)$ such that span a, where a is a subset of p, is a possible answer to q [39].

Hence, we can express the *task* of the reader as a supervised learning problem where a predictor f learns to take in a passage p and a question q and return span a as an answer [12]:

$$f : (p, q) \rightarrow a$$

3.1.2 Reading comprehension: a brief history. The 1970s saw the *QUALM* [53] system setting a strong foundation for future automated reading comprehension related-work. The computational and general complexity of the automated reading comprehension problem meant that little further progress was achieved in the field until the late 1990s.

In 1999, Hirschman et al. [32] proposed *Deep Read* - an automated reading comprehension and QA system and as well as some baseline and dataset for future reading comprehension work. This breakthrough was followed by ANLP/NAACL [1] hosting a computer-based reading comprehension workshop the subsequent year.

Over a decade later, multiple machine-learning based methods were being explored for reading comprehension that brought with them landmark datasets containing QA-pairs, and performance baselines with which to compare accuracy with others [12]. Most notably, 2015 saw researchers at *DeepMind* [30] propose a neural network (NN) model that achieved state-of-the-art accuracy as well as providing a scalable solution for the creation of large-scale training data [12]. It is argued that 2016 onward saw the QA field enter the deep-learning era [12]. During this time, Rajuparkar et al. [64] established the *SQUAD* [64] database - thought to be one of the most consequential additions to the QA field [12, 64, 74].

Since its creation, many more datasets such as *HOTPOTQA* [82], *RACE* [49], and *SQUADS's* predecessor *SQUAD 2.0* [63] have been established that have improved upon the weaknesses of *SQUAD* [12, 63].

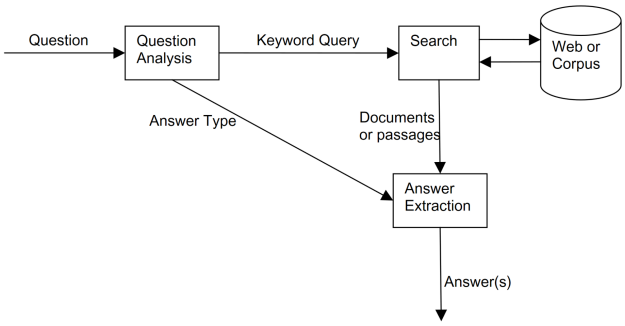

Figure 1. Architecture of QA system taken from *Open-Domain Question–Answering by John Prager* [60].

4 Knowledge-based QA

The general idea of knowledge-based QA (KB-QA) consists of mapping some natural (or partially natural) language query to a logical or semantic representation suited to query a database for an answer. Numerous datasets exist that map QA pairs with some logical form - the complexity of which vary as can be seen with the following examples [39]:

- *GeoQuery*: a dataset of question-answer pairs to US geography based factoids [27].
- *DROP*: a dataset comprising of complex questions requiring some form of reasoning to answer [22].
- *BREAK*: a dataset designed to help train a model to decompose questions via *Question Decomposition Meaning Representation* [76].

The pursuit of building a model that can generalise and capture latent structures of components of training data also garnered attention, with many models failing to capture more complex *compositional generalisation* [57]. Current literature categorises KB-QA into two main categories: *graph-based*

QA and *QA by semantic parsing* [39, 80], both requiring some form of *entity linking* prior to their execution.

4.1 Entity linking

Entity linking (EL) maps an ontological entity from a knowledge-base to some relevant textual mention [35]. This (generally) two-stage process is done by mean of *mention detection* and *mention disambiguation* [39]. Mention detection, in conjunction with *named entity recognition* (NER), identifies and classifies entities from a passage of raw text into predefined categories [81]. If a given entity refers to multiple possible classifications, mention (or *named-entity*) disambiguation seeks to map an ambiguous mention with its corresponding actual entity [33]. One such algorithm is the *TAGME* linker [23] that identifies short-phrases from within a passage of unstructured text and maps them to a relevant Wikipedia page [1]. The literature refers to such tools as *Wikification* algorithms whereby a set of entities is defined as a set of Wikipedia pages [39].

More recent EL models have utilised neural graph-based linking [39] and NN models [8] due to their impressive feature abstraction and generalisation abilities. Such models require entities to be projected as low-dimensional vectors with features of textual mentions and corresponding candidate entities being learned from data [8, 19]. Current literature shows such unsupervised learning approaches to yield performance improvements on current state-of-the-art models [19, 51].

4.2 QA by Relation Extraction

Graph-based methods model the KB as a graph with entities as nodes and propositions or relations being represented as edges. Research has shown graph-based QA methods can improve performance of information extraction, question classification, relation extraction, as well as NLP related tasks such as part-of-speech tagging and named-entity recognition [9].

4.2.1 RDF triples. In the simplest case of Graph-based QA, factoids are expressed as a set of 3-tuples containing some predicate and two arguments that serve to denote a simple proposition or relation [39]. These tuples are referred to in the literature as *Resource Description Framework* (RDF) *triples* and are deceivingly powerful and versatile forms of relation-expression [3, 72]. Within this framework, a question-task is expressed as an RDF tuple with a missing argument, where the goal of the QA system is to answers questions about *that* missing argument. Some popular examples of RDF triple databases are *DBpedia* [52] or *Freebase* [6].

To illustrate RDF based QA in action, the following example was adapted from Jurafsky [39]. Consider the following RDF triple:

`(University of Cape Town, established in, 1829).`

This can be used to answer questions such as *"When was UCT established?"* or *"What university in Cape Town was established in 1829?"*. Our next step is to determine what relation is being asked-about. Assuming entity linking has been completed, a user-query such as *"When was UCT established?"* can be mapped to some relation within the KB. Relation detection and linking can be achieved by computing some similarity metric between an encoded question and each possible relation within the KB. For the purposes of relation detection, a *BERT* model [20] can be utilised in order to encode a question's span with the probability of a particular relation. Lastly, the KB is searched for triples containing the entities and relations returned in the previous step after-which some ranking algorithm or classifier is used to determine the most likely entity-relation pair for the given question.

[1] https://tagme.d4science.org/tagme/ (Accessed May 28 2021)

4.3 QA by Semantic Parsing

Semantic parsers translate natural-language text into a *semantic representation* (or *logical form*) in a format that machines can act upon [36, 39, 42]. Within the context of QA, semantic parsers map some question to a representation that is said to be *executed* against an *environment* (typically a KB of some kind) in order to generate an answer to that question.

These formalisms can be expressed by means of predicate calculus (or *first order logic*) [69], query languages [43], a labeled graph [5], or some other executable program [71]. While it is well documented that semantic parsing acts as a boon to QA systems [54, 77, 78, 86], the extent to which supervised learning is used varies [39]. Hence, semantic parsers can be fully supervised where questions are paired with predetermined logical forms or weakly supervised, where an explicit answer in the training data is given thereby allowing for logical form to be modeled as a latent variable [39, 75].

4.3.1 Rule based. The earliest strategies for semantic parsing systems were primarily rule-based systems based heavily on pattern matching [37], or strict syntax based systems [79]. Such rule-based approaches made systems hyper-domain specific and ungeneralisable [42]. While some methods showed promise with respect to adaptability, the limitations of rule-based semantic parsers motivated researchers to look towards supervised learning.

4.3.2 Fully supervised. Fully-supervised statistical learning techniques for semantic parsing have been thoroughly explored throughout the literature [47, 85, 87]. These data-driven approaches required sentence-logical form pairs, where semantic parsing models would be trained to map the natural language sentences to a corresponding logical form. As a consequence, such systems require large amounts of annotated and compositionally-structured training data, a task that is deemed practically unfeasible for current million-scale KB or hyper-niche domains [42, 80]. Moreover, unsupervised learning for semantic parsing was crucial to address the shortcomings of data-driven approaches [59].

4.3.3 Weakly supervised. A fundamental change to the training of semantic parsers was the shift to training on the *result* of an execution of a logical form [42], where the underlying semantic representation is modeled as a latent variable. Consequently, *weak supervised learning* favours minimal domain-specific assumptions about a KB leading to greater flexibility [36]. While an improvement on previous attempts, weak supervised learning require a much larger *search-space* with which to explore during training. Such methods also fall victim to spurious correlations forming due to noise within the data [42].

4.3.4 Unsupervised. A critical challenge to supervised learning methods for semantic parsing is the high cost of creating correctly annotated data from from which to train and validate models upon [31]. Poon & Domingo [59] proposed the first approach to *unsupervised semantic parsing*. This saw tokens of the same *type* clustered together, after which expressions whose subexpressions fell within the same cluster were then recursively clustered together. Herzig & Berant [31] identify the underlying repetitive structural composition of language by training a neural semantic parser over a pooled collection of multiple KBs. Their model saw state-of-the-art accuracy on the *OVERNIGHT* dataset. The ability to generalise the underlying structure of language in order to semantically parse a KB has become a popular topic of research [42].

4.4 Encoder-decoder model

Encoder-decoder (or *sequence-to-sequence*) networks (or models) have been a topic of extensive research within recent years [42]. These networks have appeared in the literature to be applicable to a variety of NLP-based applications such as machine translation [13, 41], syntactic parsing [73], text summarising [88], question answering [18, 39], and semantic parsing [36, 42].

4.4.1 Architecture. Encoder-decoder networks are generative models capable of producing sequences of context-specific text of variable length. The fundamental idea behind this framework is the sequence-to-sequence aspect. An *encoder* network takes in a sequence as input and returns some contextualised representation as output (called the *context*). The context is passed to a *decoder* that produces some task-specific sequence as output [39].

More formally [39], an encoder is said to accept an input sequence x_1^n and generate some contextualised sequence of representations of *that* input, h_1^n. A decoder accepts a context vector c and produces a sequence of hidden states h_1^m which allow for some output states y_1^m to be obtained.

4.4.2 Recurrent neural networks. An recurrent neural network (RNN) is a neural network consisting of a hidden state h and an optional ouput y, that operates on a sequence x of variable-length T [13]. Practically, the encoder-decoder model can be applied to a pair of RNNs in order to *learn* to encode a variable-length sequence into some fixed-length representation, after which this sequence is decoded back into some new variable-length sequence. Thus, this pair of RNNs are jointly trained to maximise the conditional probability of a target sequence given a source sequence [13].

5 Interactive question-answering

Hardy et al. [29] argue that systems should strive for *conversational plausibility* and thus be modeled to imitate the nuances of natural human-to-human interaction. Within more current literature, a *dialogue system* refers to an automated system engaging in coherent dialogue with a human participant [45]. By both definition and design, a dialogue system should involve a human participant interacting with the system in order to achieve a specified goal. Hence, *interactive-question answering* (IQA) can be thought of as the union between a dialogue system and QA.

5.1 Question answering

In IQA, an agent is tasked with answering questions by interacting with a dynamic environment [26]. This environment can be in the form of a knowledge-base [21], a text-based game [15, 83], or some other kind of entity-relation schema. A primary focus of machine reading comprehension (MRC) research centers around the retrieval of *declarative knowledge* - that is, explicitly stated or static descriptions of entities within a KB [83]. Evidence suggests that current MRC and neural methods fail to fully engage in actual comprehension abilities - that is, they fail to capture an *understanding* of a required task [70, 83, 84]

5.2 User interaction

The incorporation of user feedback or utterances has been proposed as a means of improving neural semantic parsing [42]. Iyer et al. [34] proposed interactive user-feedback to improve the mapping of textual questions directly to query-language, thereby removing intermediate meaning representation. Larwence & Riezler [50] utilised user-feedback on the quality of a system, in the form of user-interaction logs, to improve neural semantic parsing. By modeling meaning representation as a latent variable, Artzi & Zettlemoyer [4] propose semantic parsers can be trained using conversational feedback.

Within recent years, there has been much work done in the ability of an agent to answer a series of interrelated questions from some KB [14, 46, 67, 68]. Reddy et al. [67] argue that conversational QA is essential for the goals of information-gathering agents while further evidence shows follow-up questions to significantly increase model accuracy [46]. Hence models should aim to comprehend the context of a question while also capturing more abstract ideas such as topic continuity and topic shift. The capturing of these latent conversational factors should allow for conversational agents to engage in more coherent, on topic, and interactive dialogue

with a user - achieving both conversational-plausibility as well as answering accuracy [46, 68].

5.3 Reinforcement Learning

Within the reinforcement learning (RL) paradigm, an agent is considered to make a sequence of policy-based decisions in order to maximise some expected reward once the sequence is finished [42]. However, due to the large action-state space of many NLP problems, there is motivation for more powerful methods of RL. Deep reinforcement learning (DRL) addressed this shortcoming of RL and has gained popularity in NLP as many related problems can be expressed as some form of sequence decision-making problem [65]. That DRL agents can attained improved accuracy over traditional methods can be trained interactively through human feedback, a growing body of research points to such agents having improved accuracy over traditional methods [56].

5.3.1 RL based QA. The embodiment hypothesis theorises intelligence to be an emergent property between an agent acting upon an environment with sensory-motor activity [38]. Das et al.'s [17] proposed *EmbodiedQA*[2] is a reinforcement learning (RL) agent trained to navigate a 3D environment in order to answer a dataset of questions. *EmbodiedQA* was trained with an explicit goal of generalising to unseen environments. Benchmark evaluations demonstrated its ability to accurately answer questions relating to the environment. Two benchmark oracles were used: one human, and one computational [17].

Literature has also shown the promise of applying RL agents to more complex QA problems [11]. Chali et al. trained an RL model to generate summaries as answers to complex questions. Evaluation results against current benchmark datasets demonstrate the effectiveness of an RL-based approach to QA. The authors also found that training procedures that allowed for real-time human interaction further improves model performance, which was corroborated by previous work [10].

Buck et al. [7] proposed an RL agent to act as an intermediary between a user and a blackbox QA system (referred to as the environment). This process sees an agent reformulates an initial question from a user into, potentially, many that are probed against the environment. The environment's responses are aggregated into a candidate 'best' answer to the initial question. These latent reformulation questions are created in order to maximise the likelihood of getting the correct answer. Evaluations found the agent to outperform a state-of-the-art base model and as well as other benchmarks. The agent was able to discover strategies for successful question-reformulation that were found to somewhat resemble established IR techniques including term re-weighting and stemming [7].

Godin et al. [25] posit that current metrics used by RL agents for QA over knowledge-graphs are inadequate for modeling results with confidence. More specifically, such systems fail to account for situations where there is, in fact, no answer within the KB to the question at hand. An RL agent was trained to answer questions relating to a knowledge graph, where it had the ability to not answer a question if it was deemed (by the agent) to not fall within the domain of the KB. Their results showed increased accuracy and a positive improvement, over previous approaches. Such findings provide evidence that abstaining from answering a question could help improve QA accuracy and effectiveness [25].

5.4 Text-based Environments

Utilising some text-based world generating framework, RL agents are able to dynamically interact with text-based environments to achieve some goal - normally rooted in information-extraction or question answering. The underlying environment with which the agent inhabits can be some abstraction of a real-world KB wherein different in-game objects or locations can be decoded as representing some real-world entity-relation pairs. Within this textual space, RL agents are expected to learn optimal policies with which some reward is maximised. These policies represent some abstraction of "skills" or strategy with a sequence of actions can maximise a reward. Such skills include certain forms of reasoning, comprehension, memory, as well as contextual-awareness of their environment [58].

One of the most popular framework for text-based games is *TextWorld* [15, 58]. TextWorld is an open-source simulator developed by *Microsoft* that aims to train RL agents to acquire skills such as decision making and language comprehension[3]. Using TextWorld, Yuan et al. developed Question Answering with Interactive Text (QAit) [83]. Here an agent interacts with a partially observable text-based environment in order to gather information such as attributes of objects such as locality, existence, and other features. With the popularity of text-based games, many of the barriers to entry for approaching RL problems in NLP are thought to have been lifted - which some posit will have a significant impact on language learning in dialogue-like environments [40, 58].

[2]https://embodiedqa.org/ (Accessed May 29 2021)

[3]https://www.microsoft.com/en-us/research/project/textworld/ (Accessed June 3 2021)

6 Conclusions

Despite showing success in earlier years, more recent NLP and QA literature has illustrated unsupervised learning to outperform standard QA procedures. One such example is the impact machine reading comprehension had on information-retrieval methods, which showed consistent state of the art improvements as more advanced ML methods were integrated into the practice. Furthermore, great success has also been observed on knowledge-based QA as the literature has shown unsupervised learning techniques to greatly improve semantic parsing. To this end, current works show sequence-to-sequence models coupled with recurrent neural networks to have demonstrated state-of-the-art aptitude over current standard practices. Such machine learning techniques aim to capture underlying structural information regarding natural language and aim to model these more abstract, latent variables with increasing success.

With such context in mind, the literature suggests that there are still a multitude of promising techniques and methods that have not yet been the subject of extensive research. Current evidence suggests that adding greater interactivity into a QA systems' training process yields significant performance improvements over traditional methods. These positive changes bring about considerable improvements with respect to a QA model's coherence, accuracy, and dialectic abilities.

A growing body of research has also indicated the novelty with which reinforcement learning-based methods are able to converge on solutions situates them well for NLP and QA problems. These techniques, while being extremely versatile in nature, also exhibit improvements over standard QA methods. The inclusion of text-based environments has also shown promise for interactive language learning problems, and when used in conjunction with RL, has demonstrated great proficiency necessary for the further research.

References

[1] 2000. *ANLP-NAACL 2000 Workshop: Conversational Systems*. https://www.aclweb.org/anthology/W00-0300

[2] Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. 2020. Learning dynamic belief graphs to generalize on text-based games. *Advances in Neural Information Processing Systems* 33 (2020).

[3] Hiba Arnaout and Shady Elbassuoni. 2018. Effective searching of RDF knowledge graphs. *Journal of Web Semantics* 48 (2018), 66–84.

[4] Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 421–432.

[5] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*. 178–186.

[6] Kurt Bollacker, Robert Cook, and Patrick Tufts. 2007. Freebase: A shared database of structured general human knowledge. In *AAAI*, Vol. 7. 1962–1963.

[7] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830* (2017).

[8] Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. *arXiv preprint arXiv:1811.08603* (2018).

[9] Asli Celikyilmaz, Marcus Thint, and Zhiheng Huang. 2009. A graph-based semi-supervised learning for question-answering. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 719–727.

[10] Yllias Chali, Sadid A Hasan, and Kaisar Imam. 2012. Improving the performance of the reinforcement learning model for answering complex questions. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2499–2502.

[11] Yllias Chali, Sadid A Hasan, and Mustapha Mojahid. 2015. A reinforcement learning formulation to the complex question answering problem. *Information Processing & Management* 51, 3 (2015), 252–272.

[12] Danqi Chen. 2018. *Neural reading comprehension and beyond*. Stanford University.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[14] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).

[15] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*. Springer, 41–75.

[16] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2019. KBQA: learning question answering over QA corpora and knowledge bases. *arXiv preprint arXiv:1903.02419* (2019).

[17] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–10.

[18] Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7651–7658.

[19] Ziheng Deng, Zhixu Li, Qiang Yang, Qingsheng Liu, and Zhigang Chen. 2020. Improving Entity Linking with Graph Networks. In *International Conference on Web Information Systems Engineering*. Springer, 343–354.

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

[21] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777* (2016).

[22] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161* (2019).

[23] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). *CoRR* abs/1006.3498, 1625–1628. https://doi.org/10.1145/1871437.1871689

[24] Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Christian Jacquemin, Laura Monceaux, and A Vilnat. 2002. How NLP can improve question answering. *KO Knowledge Organization* 29, 3-4 (2002), 135–155.

[25] Fréderic Godin, Anjishnu Kumar, and Arpit Mittal. 2019. Learning when not to answer: a ternary reward structure for reinforcement learning based question answering. *arXiv preprint arXiv:1902.10236* (2019).

[26] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4089–4098.

[27] Arthur C Graesser, Cathy L McMahen, and Brenda K Johnson. 1994. Question asking and answering. (1994).

[28] Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications* 53, 4 (2012).

[29] Hilda Hardy, Alan Biermann, R Bryce Inouye, Ashley McKenzie, Tomek Strzalkowski, Cristian Ursu, Nick Webb, and Min Wu. 2006. The Amitiés system: Data-driven techniques for automated dialogue. *Speech Communication* 48, 3-4 (2006), 354–373.

[30] Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340* (2015).

[31] Jonathan Herzig and Jonathan Berant. 2017. Neural semantic parsing over multiple knowledge-bases. *arXiv preprint arXiv:1702.01569* (2017).

[32] Lynette Hirschman, Marc Light, Eric Breck, and John D Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*. 325–332.

[33] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 782–792.

[34] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. *arXiv preprint arXiv:1704.08760* (2017).

[35] Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 1148–1158.

[36] Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622* (2016).

[37] Tim Johnson. 1984. Natural language computing: the commercial applications. *The Knowledge Engineering Review* 1, 3 (1984), 11–23.

[38] Dale W Jorgenson. 1966. The embodiment hypothesis. *Journal of Political Economy* 74, 1 (1966), 1–17.

[39] Dan Jurafsky. 2020. *Speech and language processing*. preprint on webpage at https://web.stanford.edu/~jurafsky/slp3/ed3book_dec302020.pdf.

[40] Dan Jurafsky and James H Martin. 2014. Speech and language processing. Vol. 3. *US: Prentice Hall* (2014).

[41] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1700–1709.

[42] Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978* (2018).

[43] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, and Hongrae Lee. 2020. Natural language to SQL: Where are we today? *Proceedings of the VLDB Endowment* 13, 10 (2020), 1737–1750.

[44] Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. *Information Sciences* 181, 24 (2011), 5412–5434.

[45] Natalia Konstantinova and Constantin Orasan. 2013. Interactive question answering. In *Emerging applications of natural language processing: concepts and new research*. IGI Global, 149–169.

[46] Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to Identify Follow-Up Questions in Conversational Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 959–968.

[47] Tom Kwiatkowksi, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. 1223–1233.

[48] Cody CT Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. In *Proceedings of the 10th international conference on World Wide Web*. 150–161.

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

[50] Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. *arXiv preprint arXiv:1805.01252* (2018).

[51] Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. *arXiv preprint arXiv:1804.10637* (2018).

[52] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.

[53] Wendy G Lehnert. 1977. A conceptual theory of question answering. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*. 158–164.

[54] Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics* 39, 2 (2013), 389–446.

[55] Diego Mollá, Menno Van Zaanen, Daniel Smith, et al. 2006. Named entity recognition for question answering. (2006).

[56] Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. *arXiv preprint arXiv:1603.07954* (2016).

[57] Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. Improving compositional generalization in semantic parsing. *arXiv preprint arXiv:2010.05647* (2020).

[58] Tatiana-Andreea Petrache, Traian Rebedea, and Ştefan Trăuşan-Matu. [n.d.]. Interactive language learning-How to explore complex environments using natural language? ([n. d.]).

[59] Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 conference on empirical methods in natural language processing*. 1–10.

[60] John M Prager. 2006. Open-Domain Question-Answering. *Found. Trends Inf. Retr.* 1, 2 (2006), 91–231.

[61] Peng Qi, Haejun Lee, Oghenetegiri Sido, Christopher D Manning, et al. 2020. Retrieve, rerank, read, then iterate: Answering open-domain questions of arbitrary complexity from text. *arXiv preprint arXiv:2010.12527* (2020).

[62] Silvia Quarteroni and Suresh Manandhar. 2009. Designing an interactive open-domain question answering system. *Natural Language Engineering* 15, 1 (2009), 73.

[63] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).

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

[65] Rajkumar Ramamurthy, Rafet Sifa, and Christian Bauckhage. 2020. NLPGym–A toolkit for evaluating RL agents on Natural Language Processing Tasks. *arXiv preprint arXiv:2011.08272* (2020).

[66] Rami Reddy, Nandi Reddy, and Sivaji Bandyopadhyay. 2006. Dialogue based question answering system in telugu. In *Proceedings of the Workshop on Multilingual Question Answering-MLQA '06*.

[67] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[68] Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[69] Erik J Sandewall. 1970. *Representing natural-language information in predicate calculus*. Technical Report. STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE.

[70] Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? *arXiv preprint arXiv:1808.09384* (2018).

[71] Maarten H Van Emden and Robert A Kowalski. 1976. The semantics of predicate logic as a programming language. *Journal of the ACM (JACM)* 23, 4 (1976), 733–742.

[72] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. *Journal of Web Semantics* 37 (2016), 184–206.

[73] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[74] Soumya Wadhwa, Khyathi Raghavi Chandu, and Eric Nyberg. 2018. Comparative analysis of neural QA models on squad. *arXiv preprint arXiv:1806.06972* (2018).

[75] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. 2016. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence* 39, 11 (2016), 2314–2320.

[76] Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics* 8 (2020), 183–198.

[77] Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 439–446.

[78] Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 960–967.

[79] William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*. 441–450.

[80] Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957* (2016).

[81] Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1237–1247.

[82] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 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, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259

[83] Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Christopher Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive language learning by question answering. *arXiv preprint arXiv:1908.10909* (2019).

[84] Xingdi Yuan, Jie Fu, Marc-Alexandre Cote, Yi Tay, Christopher Pal, and Adam Trischler. 2019. Interactive machine comprehension with information seeking agents. *arXiv preprint arXiv:1908.10449* (2019).

[85] John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*. 1050–1055.

[86] Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. 678–687.

[87] Luke S Zettlemoyer and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420* (2012).

[88] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. *arXiv preprint arXiv:1704.07073* (2017).