Interactive Question Answering

Literature Review

Roy Cohen
CHNROY002@myuct.ac.za
UCT Computer Science Honours

Abstract

This literature review addresses the task of Interactive Question Answering (IQA) over a Knowledge Base (KB) and will examine the core principles of Question Answering (QA) and the various techniques used to improve upon it. QA systems are a popular and powerful tool for retrieving answers from structured databases. Different state-of-the-art techniques used in QA such Natural Language Processing and Reinforcement Learning (RL) are investigated and compared according to their strengths and limitations as outlined by the literature. Finally, this paper examines Knowledge Bases and text-based environments for their viability in producing accurate and robust IQA systems. The investigation finds that problems of IQA can be framed an RL problem, prompting further research into RL based IQA as a generalised tool to answer complex questions about a KB.

1 Introduction

NLP is a computer motivated approach to analysing and representing natural language that occurs in text such that multiple different linguistic analyses can be done to achieve human-like language processing [16]. *Question Answering* (QA), which lies at the intersection of *Information Extraction* and NLP [34], has consistently been an area of active research.

QA systems are a powerful tool for answering questions in natural language [3, 13, 21]. Throughout the past decade, demand for QA systems that are accurate, efficient, and cost effective to train have only been increasing. The ubiquity brought upon by recent advances in machine-learning and Natural Language Processing (NLP) has served to further bolster the QA research community prompting further investigation into the field [54, 65].

Furthermore, with the surge in popularity of intelligent personal assistants, such as Alexa and Siri, research has also been focused on *Interactive Question Answering* (IQA), more specifically *Conversational Question Answering* (ConvQA) which allows for dialog to take place between a user and the system [35],

This literature review will explore literature revolving around state-of-the-art Knowledge Base Question Answering (KB-QA) systems using reinforcement learning [19, 26, 36, 72] and NLP techniques such as semantic parsing [7, 23] and embedding [9, 12, 74].

Furthermore, an interactive text-based game environment, TextWorld [17], will be reviewed as a means to develop state-of-the-art Reinforcement Learning based QA system [79].

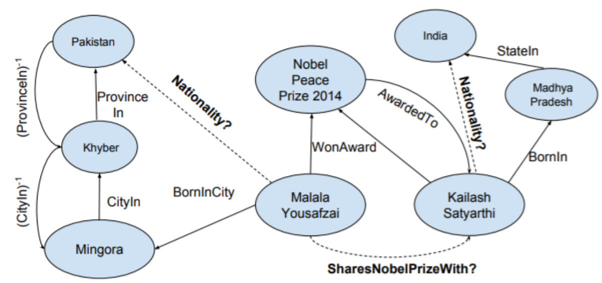

Figure 1. A small piece of a KB represented as a KG, from Das et al. "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning" [19]. Solid edges are KB relations and dotted lines are queries.

2 Knowledge Bases

Knowledge Bases (KB) are frequently used as a structured database that stores data in the form of tuples. Tuples are in the form of `(entity_head, relation, entity_tail)` [10, 22, 77] and natural language questions are typically translated into a KB query in the form of `(entity_head, relation, ?)`. Where the tail entity would be the answer to the question. An example of this would be `(Cape Town, cityIn, South Africa)`.

Since the introduction of free large-scale tuple based KBs, such as *Freebase* [8], and *DBPedia* [4], they have been extensively used as an open-domain source of information for many different NLP methods such as entity-linkers, relation extraction, and question answering systems [35, 57].

KBs have two properties, namely *heterogeneity* and *imbalance* [31]. Heterogeneity is the property that relations can link many entity pairs while others do not. The imbalance property states that the number of head entities in a relation may differ from the number of tail entities.

2.1 Knowledge Graph

A Knowledge Graph (KG) is a graph based representation of a KB, which is created by treating *entities* as nodes and *relations* as edges, which is illustrated in Figure 1.

2.2 Entity Linking

Entity linking is the process of matching a natural-language mention with the representation of an entity in an ontology (e.g. KB) [32]. Given a KB which has two instances of the same entity (such as two people with the same name), it is the task of entity linker to decide which entity is being referred to in the question.

Entity linking can usually be broken down into two parts, *mention detection* and *entity disambiguation* [39]. Mention detection is the extracting of references to natural language entities, while entity disambiguation is tasked with matching mentions to their corresponding entities.

A popular entity linking algorithm used as a baseline is the *TAGME* linker [24]. The *TAGME* algorithm handles both mention detection and entity disambiguation [34].

Entity linking models based on deep learning have also managed to perform with state-of-the-art results [56], and have shown major improvements over traditional approaches [14, 45, 55], since traditional approaches cannot encapsulate all statistical dependencies, and relations [25].

Extracting a mention of an entity and extracting the correct mention of that entity from a knowledge base is critical for retrieving information accurately from and knowledge base, and allows accurate retrieval of entities in question answering systems.

3 Question Answering

Knowledge Base Question Answering systems (KB-QA) are used as a means to convert a user's natural language question to a Knowledge Graph query which retrieves the correct answer from information in KB tuples [5, 6, 10, 58, 73, 75, 76].

Dwivedi et al. [21] state that the broad stages of QA systems include firstly *question analysis*, which includes parsing of natural language, classification of the question and the possible reformulation of the query. The next stage is *document analysis* where candidate documents are retrieved, and potential answers are identified. The final stage, *answer analysis*, extracts prospect answers and gives them some ranking to identify the best answer.

Questions can be classified according the answer they are expected to produce. The types of questions are *factoid*, *list*, *explanation* and *complex questions* [40].

- Factoid questions are those with which the answer is a simple fact, answerable in a few words [29]. Example: *Who is the president of South Africa?*.
- List questions have a set of answers that all satisfy the question [29]. Example: *Who has played James Bond?*.
- Definition questions are those where the answer required a short paragraph explanation [52]. Example: *How does a CPU work?*.
- Complex questions require multiple steps of reasoning [36]. Example: Figure 1, shows how the question "What is Malala Yousafzai nationality?". The answer can be found through multiple steps of logic through the KG (also known as *multi-hop*), but not through a single KG relation.

$question_1$: When was Avengers: Endgame released in Germany?
$answer_1$: April 2019
$question_2$: What was the next from Marvel?
$answer_2$ Spider-Man: Far from Home
$question_3$: Released on?
$answer_3$: July 2019
$question_4$: So who was Spidey?
$answer_4$: Tom Holland
$question_5$: And his girlfriend was played by?
$answer_5$: Zendaya Coleman An ideal conversation, from (Kaiser et al., 2021)

Figure 2. An ideal conversation, from Kaiser et al. "Reinforcement Learning from Reformulations in Conversational QuestionAnswering over Knowledge Graphs" [36]

3.1 Interactive QA

Interactive QA (IQA) is a combination of question answering systems and dialog systems, where question answering allow users to ask questions in natural language and receive answers, while dialog systems allows the system to exchange in dialog with the user in cases where there are multiple answers, no answers, or there is ambiguity in the question being asked. [41].

3.1.1 Conversational QA. Conversational QA (ConvQA) systems are a type of IQA, which work in a consecutive multi-turn informational retrieval and have become a suitable mechanism to retrieve information that cannot be retrieved through a single logical path in a structured database (in our case, a KB) [36]. ConvQA is traditionally achieved through semantic parsing solutions, which convert natural language into KB queries to extract answers [27, 58, 80].

Figure 2 shows a snippet, from Kaiser et al. [36], that shows an ideal conversation with five turns. Figure 2 also shows challenges faced in QA systems such as textual mentions of words can be colloquial (Figure 2 $question_4$), incomplete (Figure 2 $question_2$ & $question_3$) and question context can change in ConvQA systems (Figure 2 $question_5$) [36].

4 Natural Language Processing

NLP is a computer motivated approach to analysing and representing natural language to a near human-like level [16]. In order to achieve near human-like language processing, it is critical to construct a syntactic structure to analyse [62]. The construction of syntactic structures is referred to as parsing.

4.1 Semantic Parsers

Semantic parsing is used to map natural language to a formal meaning representation [53], as seen in Figure 3. This meaning representation can be expressed in logical forms, such

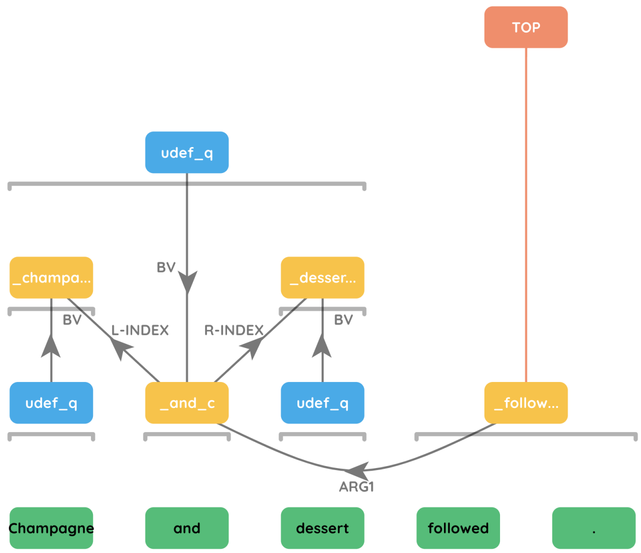

Figure 3. Visual representation of a meaning representation graph

as predicate calculus or SQL, where it can be treated as an executable program [34].

Early Semantic parsers were *rule based* and were traditionally constructed manually. Rule based parsing was a domain specific semantic parsing method [38]. These early attempts utilised pattern matching [33], syntax-based systems [69]. While rule based approaches were relatively simple, they were domain specific and thus not adaptable to other domains.

Many *fully supervised* machine learning methods proposed [50, 82] uses a fully annotated data-set of sentences and their corresponding logical expressions as training data [42]. A big limitation on fully supervised machine learning methods these types of machine learning methods is that they require large amounts of high-quality annotations of example sentences to train on, which can require non-trivial efforts.

Weakly supervised methods [6] have also been suggested, by using a data-set of question and answers as training data. However weakly supervised method require larger *search-space* to train on than other methods, and also get affected by high noise contained in the data which can result in two unrelated variables to be correlated [38].

Due to the limitation of both fully and weak supervision methods, *unsupervised methods* [53] have also been proposed which by clustering tokens with the same type using the clusters to combine sub-expressions [38].

4.2 Semantic Parsing Based KB-QA

QA using semantic parsers are done by mapping the natural language question to a formal meaning representation [34], which can be treated as an executable program, then queries on the KB and extracts the answer.

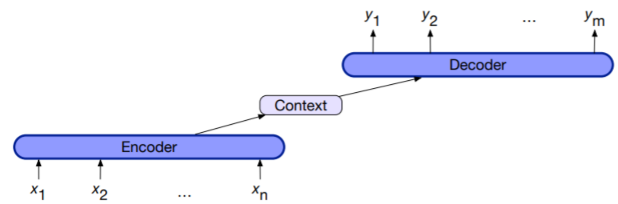

Figure 4. Sequence to Sequence model architecture, from Jurafsky "Speech and language processing" [34].

Despite semantic parsers' success in literature [43, 68, 80, 81], it has been argued that semantic parsing based approaches to QA do not scale well [7].

4.3 Sequence to Sequence Models

Using Neural Networks for semantic parsing have been produced underwhelming results [47, 80, 82].

Liang et al, in 2017, introduced Neural Symbol Machine (NSM) [46], which outperformed state-of-the-art semantic parsers. This was achieved through the use of sequence to sequence models.

Sequence to Sequence models, are models that manage to compute contextual output sequence that isn't limited by output length. Many applications exist for sequence to sequence models such as semantic parsing [47], syntactic parsing [63], image captioning [64], machine translation [15, 37], etc.

The underlying architecture is to have an encoder network that translates natural language into a contextualisation representation of it (the *context vector*) [34]. The *context vector* is then passed onto the decoder, which outputs a sequence of tokens. Figure 4 shows a visual representation of the sequence to sequence model.

Liang et al. [46] used a *Recurrent Neural Network* (RNN) as both the encoder and decoder proposed by Cho et al. [15], in their sequence to sequence model. RNNs is a neural network that contains cycles [34], allowing for arbitrary length input and output. Thus the encoder-decoder RNNs are trained together to learn to encode any length natural language input into a set length *context vector*, and decoded back into a sequence of tokens [46].

Using sequence-to-sequence model approach to IQA could result in state-of-the-art performance just as it results in state-of-the-art semantic parsing.

4.4 Embedding Based KB-QA

Embedding is the task of translating natural language questions to vectorial representations [12]. The vectorial representations do not require pre-defined grammars or lexicons and can retrieve information from KB independent from its schema [12].

Embedding based models can be split up into two parts, the encoder and the decoder. The encoder is tasked with summarizing the natural language input into the low dimensional vector representation, while the decoder generated the output from the low dimensional vector representation [78].

Embedding models have managed to achieve performance close to state-of-the-art, such as TransE [11], whose simple and effective embedding model achieves state-of-the-art prediction performance. However, TransE struggles with dealing with one to many, many to one, and many to many relations in KBs [30]. TransH [66] attempts to fix these issues by translating vectors on a hyperplane.

However, both TransE and TransH assume that entities and relations in KB tuples share the same vector space [30]. Lin et al. [48] proposed TransR and CTransR based on the idea that entities and relations are different objects. However, the drawback of this is that the model is computationally expensive and cannot be applied to large-scale KBs [30].

Ji et al. [30], in 2015, took this idea a step further and proposed a more fine-grained model, TransD. This model used fewer parameters and included no matrix-vector multiplications (only vector operations), allowing the TransD to scale to larger KBs, with better performance than TransR and CTransR.

All embedding methods mentioned so far ignore the two properties of KBs (*heterogeneity* and *imbalance*) [31]. Lack of the heterogeneity property will cause complex questions to be under fitted or simple factoids to be overfitted [18]. The imbalance property suggests that head and tail entities should be treated differently [18].

TransParse [31], TransA [70], and TransG [71] provides an embedding model that acknowledges KBs properties, while TransA and TransG continue to improve performance by introducing different Gaussian mixture models to embed entities and relations [18].

Embedding based QA systems have the limitation that they find capturing complex reasoning (which is needed to answer complex questions) challenging. There have been approaches attempting to overcome this limitation by performing random walks around the KG [28, 44, 51, 60], independent of the query relation [19]. This leads to inefficiency since random paths need to be pre-computed.

This is one of the motivations for using a Reinforcement Learning based approach to KB-QA.

5 Reinforcement Learning

Reinforcement Learning (RL) is often considered as the third machine learning paradigm (with supervised and unsupervised being the other two), where an agent in an environment is learning what to do by mapping situations to different actions and doing so to maximize some reward signal it received for performing these actions [59].

There are three key categories of reinforcement learning, namely *dynamic programming*, *Monte Carlo* methods and *temporal difference* methods [61]. All three categories frame the decision-making problem into a Markov Decision Process (MDP), a mathematics-based method for solving decision-making problems with four key elements.

5.1 Elements of MDP

A *policy* dictates the agent's way of making decisions by mapping a set of states in the environment observed by the agent to actions which it will take [59]. Generally, policies can be stochastic by providing a probability for each action.

A *reward signal* is used to define a reinforcement problem's goal by providing a reward to the agent for making decisions (i.e. actions which will lead the agent closer to its goal will have more rewards than ones that do not). Its the objective of the agent to make decisions that will maximize its rewards over the long run. Furthermore, it is the reward function that will motivate a change in an agents policy. If a policy's action leads to low rewards, then the policy may be altered [59].

While the reward signal is used to indicate immediate rewards after an action is taken, the *value function* is used to specify what action is good in the long run [59]. In other words, it can be seen as the total amount of reward expected from the agent over multiple time steps. The value function determines the long term effect of making an action by considering the available actions and rewards in the next state [59].

The last element is the *model*, which is used to make inference about what is accessible to the agent about the actions in the environment. This allows an agent *plan* for future states and their rewards. This element is optional, which means RL algorithms that don't use a model are called *model-free*, whereas ones with models are called *model-based* [59].

5.2 RL based KB-QA

The use of RL for KB-QA has recently become a more popular approach to QA systems. Recent literature [19, 26, 36, 72] show that using RL in QA systems manage to achieve state-of-the-art results, even with complex questions (unlike embedding and semantic parsing based methods) that need multiple steps of reasoning. Das et al. and Xiong et al. [19, 72] use KGs as an environment for RL agents to traverse and learn question answering.

In 2017, Xiong et al. [72], proposed an RL algorithm called DeepPath, which learns to pick paths between entity pairs in KGs. DeepPath had the limitation of needing to know the answer entity in advance, making their agent dependent on knowing the answer entity.

To overcome this limitation, Das et al. [19] introduced *MINERVA*, which uses *policy gradients* to learns how to answer queries by traversing a Knowledge Graph, which managed to outperform DeepPath without needing to know the answer entity.

More recently, Kaiser et al. [36] developed a state-of-the-art RL agent called *CONQUER*, which is a model-free algorithm that is trained with a policy gradient algorithm called *REINFORCE* [67] to answer questions in a ConvQA system.

It is done by training the agent to reformulate the question being asked to find the best answer.

Godin et al. [26], in 2019, took a different approach to ConvQA, in that they trained an agent using *REINFORCE* to not only walk on a Knowledge Graph and come to the correct answer but also know when not to give an answer if it is unsure. Godein et al. [26] also argued that current metrics lack, as they do not take into consideration situations where the KB does not have the answer. Thus a new metric was proposed where the testing data included examples where there was no answer, and the QA system must report so. Using this new metric *CONQUER* outperformed *MINERVA*.

The last two agents use the *REINFORCE* algorithm, which has the disadvantage high variance and a lack of general converge theory, which makes the policy susceptible to a false maximum [67].

5.3 TextWorld

In 2019, Côté et al. [17], introduced TextWorld, which is a Python library that provides an interactive text-based environment for RL agents to learn and train. More specifically, it creates a framework in which interactive text-based games can be developed, along with question-answer pairs. As a text-based game, TextWorld can be used to research and develop more generalised QA systems [1].

The interactive environment created is based on a provided KB, where entities and relations translate to locations and actions in TextWorld, allowing agents to roam the environment and maximize their rewards by learning the optimal policy.

Using TextWorld, Yuan et al. [79] showed that an agent tasked with interacting in a partially observable text-based environment in order to extract information posed challenges.

Yuan et al. [79] suggested improvements to the model such as *structured memory* [2, 20] to allow an agent to recall locations they have already explored. Another problem faces were model overfitting [79], which could be resolved by implementing *intermediate rewards* to sub-tasks, which would reward the agent at intermediate steps when verifying attributes (answers) that require long procedures to solve [79], such as complex questions.

Another improvement could be to replace the *valued-based* methods used by Yuan et al. [79], which is argued to have poor convergence [49].

6 Conclusions

Natural language is inherently paradoxical, ambiguous, and inconsistent. This makes traditional NLP approaches to QA challenging. Hence, this paper outlines significant advances in QA using state-of-the-art NLP methods such as semantic parsing, embedding, and sequence to sequence models to address these structural and syntactic problems.

The literature indicates that many NLP QA models have a structural and syntactic understanding of sentence construction but fail to understand natural language. As seen through many widely used QA solutions that fail to comprehend complex reasoning about a knowledge base. This is motivation to make question answering systems interactive, which allows us to frame problems of IQA as RL problems.

Literature shows that RL based QA systems are better able to reason over complex chains of reasoning than traditional QA systems.

However, current RL based IQA systems have found the ability to generalise to unseen environments challenging. This strengthens the notion to continue research on RL based IQA systems to answer complex questions in generalised environments and possibly using an hybrid of traditional NLP based approaches and RL methods could lead to better performing and more accurate IQA systems.

References

[1] Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L. Hamilton. 2021. Learning Dynamic Belief Graphs to Generalize on Text-Based Games. arXiv:2002.09127 [cs.CL]

[2] Prithviraj Ammanabrolu and Mark O. Riedl. 2019. Playing Text-Adventure Games with Graph-Based Deep Reinforcement Learning. arXiv:1812.01628 [cs.CL]

[3] Ahlam Ansari, Moonish Maknojia, and Altamash Shaikh. 2016. Intelligent question answering system based on artificial neural network. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*. IEEE, 758–763.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.

[5] Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1431–1440.

[6] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 1533–1544. https://www.aclweb.org/anthology/D13-1160

[7] Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1415–1425. https://doi.org/10.3115/v1/P14-1133

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[9] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. arXiv:1406.3676 [cs.CL]

[10] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. arXiv:1506.02075 [cs.LG]

[11] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

[12] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open Question Answering with Weakly Supervised Embedding Models. arXiv:1404.4326 [cs.CL]

[13] Yllias Chali, Sadid A Hasan, and Shafiq R Joty. 2011. Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels. *Information Processing & Management* 47, 6 (2011), 843–855.

[14] Andrew Chisholm and Ben Hachey. 2015. Entity Disambiguation with Web Links. *Transactions of the Association for Computational Linguistics* 3 (2015), 145–156. https://doi.org/10.1162/tacl_a_00129

[15] 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).

[16] Gobinda G. Chowdhury. 2003. Natural language processing. *Annual Review of Information Science and Technology* 37, 1 (2003), 51–89. https://doi.org/10.1002/aris.1440370103 arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/aris.1440370103

[17] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. TextWorld: A Learning Environment for Text-based Games. arXiv:1806.11532 [cs.LG]

[18] Yuanfei Dai, Shiping Wang, Neal N. Xiong, and Wenzhong Guo. 2020. A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics* 9, 5 (2020). https://doi.org/10.3390/electronics9050750

[19] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851* (2017).

[20] Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. arXiv:1810.05682 [cs.CL]

[21] Sanjay K Dwivedi and Vaishali Singh. 2013. Research and reviews in question answering system. *Procedia Technology* 10 (2013), 417–424.

[22] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1608–1618.

[23] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering over Curated and Extracted Knowledge Bases *(KDD '14)*. Association for Computing Machinery, New York, NY, USA, 1156–1165. https://doi.org/10.1145/2623330.2623677

[24] Paolo Ferragina and Ugo Scaiella. 2010. Fast and accurate annotation of short texts with Wikipedia pages. arXiv:1006.3498 [cs.IR]

[25] Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2619–2629. https://doi.org/10.18653/v1/D17-1277

[26] 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).

[27] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-Action: Conversational Question Answering Over a Large-Scale Knowledge Base. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/d63fbf8c3173730f82b150c5ef38b8ff-Paper.pdf

[28] Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing Knowledge Graphs in Vector Space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 318–327. https://doi.org/10.18653/v1/D15-1038

[29] Matthias H Heie, Edward WD Whittaker, and Sadaoki Furui. 2012. Question answering using statistical language modelling. *Computer Speech & Language* 26, 3 (2012), 193–209.

[30] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[31] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, Vol. 30.

[32] Heng Ji and Ralph Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. 1148–1158.

[33] Tim Johnson. 1984. Natural Language Computing: The Commercial Applications. *The Knowledge Engineering Review* 1, 3 (1984), 11–23. https://doi.org/10.1017/S0269888900000588

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

[35] Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational Question Answering over Knowledge Graphs with Transformer and Graph Attention Networks. arXiv:2104.01569 [cs.CL]

[36] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Reinforcement Learning from Reformulations in Conversational Question Answering over Knowledge Graphs. *CoRR* abs/2105.04850 (2021). arXiv:2105.04850 https://arxiv.org/abs/2105.04850

[37] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 1700–1709. https://www.aclweb.org/anthology/D13-1176

[38] Aishwarya Kamath and Rajarshi Das. 2019. A Survey on Semantic Parsing. arXiv:1812.00978 [cs.CL]

[39] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-End Neural Entity Linking. arXiv:1808.07699 [cs.CL]

[40] 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.

[41] Natalia Konstantinova and Constantin Orasan. 2013. *Interactive Question Answering*. 149 –. https://doi.org/10.4018/978-1-4666-2169-5.ch007

[42] Pawan Kumar and Srikanta Bedathur. 2020. A Survey on Semantic Parsing from the perspective of Compositionality. arXiv:2009.14116 [cs.CL]

[43] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification *(EMNLP '10)*. Association for Computational Linguistics, USA, 1223–1233.

[44] Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., 529–539. https://www.aclweb.org/anthology/D11-1049

[45] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A Selective Context Model for Entity Resolution. *Transactions of the Association for Computational Linguistics* 3 (2015), 503–515. https://doi.org/10.1162/tacl_a_00154

[46] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 23–33. https://doi.org/10.18653/v1/P17-1003

[47] Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning Dependency-Based Compositional Semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 590–599. https://www.aclweb.org/anthology/P11-1060

[48] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[49] Siqi Liu, Kee Yuan Ngiam, and Mengling Feng. 2019. Deep Reinforcement Learning for Clinical Decision Support: A Brief Survey. arXiv:1907.09475 [cs.LG]

[50] Raymond J Mooney. 2007. Learning for semantic parsing. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 311–324.

[51] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional Vector Space Models for Knowledge Base Completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 156–166. https://doi.org/10.3115/v1/P15-1016

[52] Mariana Neves and Ulf Leser. 2015. Question answering for biology. *Methods* 74 (2015), 36–46.

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

[54] Sameerchand Pudaruth, Kajal Boodhoo, and Lushika Goolbudun. 2016. An intelligent question answering system for ict. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, 2895–2899.

[55] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia *(HLT '11)*. Association for Computational Linguistics, USA, 10 pages.

[56] Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2021. Neural Entity Linking: A Survey of Models Based on Deep Learning. arXiv:2006.00575 [cs.CL]

[57] Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-Task Learning for Conversational Question Answering over a Large-Scale Knowledge Base. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2442–2451. https://doi.org/10.18653/v1/D19-1248

[58] Kuldeep Singh, Arun Sethupat Radhakrishna, Andreas Both, Saeedeh Shekarpour, Ioanna Lytra, Ricardo Usbeck, Akhilesh Vyas, Akmal Khikmatullaev, Dharmen Punjani, Christoph Lange, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2018. Why Reinvent the Wheel: Let's Build Question Answering Systems Together *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1247–1256. https://doi.org/10.1145/3178876.3186023

[59] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[60] Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional Learning of Embeddings for Relation Paths in Knowledge Base and Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1434–1444. https://doi.org/10.18653/v1/P16-1136

[61] Victor Uc-Cetina, Nicolás Navarro-Guerrero, Anabel Martin-Gonzalez, Cornelius Weber, and Stefan Wermter. 2021. Survey on reinforcement learning for language processing.

[62] Roger PG Van Gompel and Martin J Pickering. 2007. Syntactic parsing. *The Oxford handbook of psycholinguistics* (2007), 289–307.

[63] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a Foreign Language *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 9 pages.

[64] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. arXiv:1411.4555 [cs.CV]

[65] Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 200–207.

[66] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

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

[68] 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*. Association for Computational Linguistics, Prague, Czech Republic, 960–967. https://www.aclweb.org/anthology/P07-1121

[69] W. 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* (New York, New York) *(AFIPS '73)*. Association for Computing Machinery, New York, NY, USA, 441–450. https://doi.org/10.1145/1499586.1499695

[70] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. 2015. TransA: An Adaptive Approach for Knowledge Graph Embedding. arXiv:1509.05490 [cs.CL]

[71] Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. TransG : A Generative Model for Knowledge Graph Embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 2316–2325. https://doi.org/10.18653/v1/P16-1219

[72] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2018. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. arXiv:1707.06690 [cs.CL]

[73] 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).

[74] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. 2014. Joint Relational Embeddings for Knowledge-based Question Answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 645–650. https://doi.org/10.3115/v1/D14-1071

[75] Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 956–966.

[76] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. (2015).

[77] Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 643–648.

[78] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural Generative Question Answering. arXiv:1512.01337 [cs.CL]

[79] Xingdi Yuan, Marc-Alexandre Cote, Jie Fu, Zhouhan Lin, Christopher Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive Language Learning by Question Answering. arXiv:1908.10909 [cs.CL]

[80] 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.

[81] Luke S. Zettlemoyer and Michael Collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (Edinburgh, Scotland) *(UAI'05)*. AUAI Press, Arlington, Virginia, USA, 658–666.

[82] Luke S. Zettlemoyer and Michael Collins. 2012. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. arXiv:1207.1420 [cs.CL]