A Survey of Knowledge Graph Based Question Answering

Cheng Huang
huangcheng18s@ict.ac.cn

August 2019

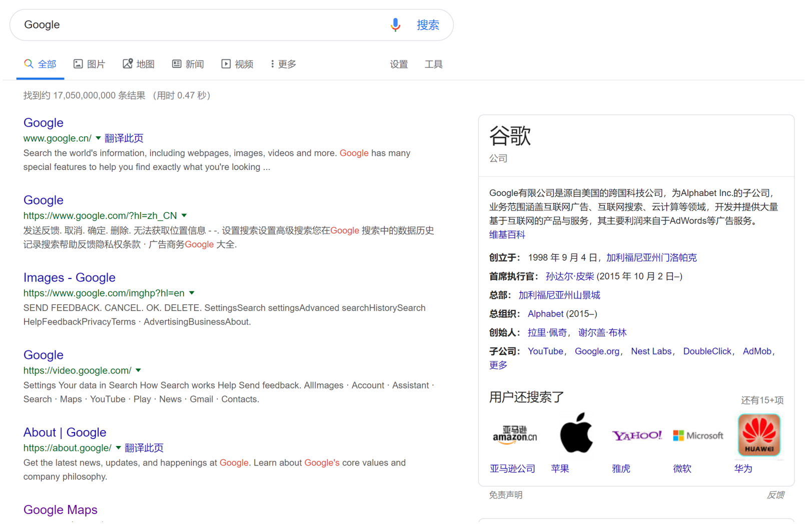
Abstract

Knowledge based system is one of the major fields in AI research. Knowledge graph based question answering system is a production that has a great outlook. There are two phases in question answering: transferring natural language to a standard format, then using this format to search for the answer. In this article, we mainly introduce different techniques to implement those two procedures.

1 Introduction

Turing test is one of the most famous concepts in artificial intelligence. According to Wikipedia [5], Turing test is a test of machine's ability to exhibit intelligent (or indistinguishable from human's) behavior. A human evaluator has natural language conversations with a human and a machine in text, respectively. If the evaluator can't tell which one is with the machine, we say the machine passes Turing test, thus it is a (strong) AI.

Nowadays, when talking about AI, we usually indicate something about deep learning. There are actually several different approaches to AI. One of those approaches, maybe the most intuitive one, is knowledge-based system. Suppose a machine has a large knowledge database and the ability of comprehending and inference, it can then pass Turing test without question. After all, in a macro perspective, that's what we humans do when taking such a test.



找到约 17,050,000,000 条结果 （用时 0.47 秒）

Google
www.google.cn/ ▾ 翻译此页
Search the world's information, including webpages, images, videos and more. Google has many special features to help you find exactly what you're looking ...

Google
https://www.google.com/?hl=zh_CN ▾
发送反馈. 取消. 确定. 删除. 无法获取位置信息 - -. 设置搜索设置高级搜索您在Google 搜索中的数据历史记录搜索帮助反馈隐私权条款 · 广告商务Google 大全.

Images - Google
https://www.google.com/imghp?hl=en ▾
SEND FEEDBACK. CANCEL. OK. DELETE. SettingsSearch settingsAdvanced searchHistorySearch HelpFeedbackPrivacyTerms · AdvertisingBusinessAbout.

Google
https://video.google.com/ ▾
Settings Your data in Search How Search works Help Send feedback. AllImages · Account · Assistant · Search · Maps · YouTube · Play · News · Gmail · Contacts.

About | Google
https://about.google/ ▾ 翻译此页
Get the latest news, updates, and happenings at Google. Learn about Google's core values and company philosophy.

Google Maps

谷歌
公司

Google有限公司是源自美国的跨国科技公司，为Alphabet Inc.的子公司，业务范围涵盖互联网广告、互联网搜索、云计算等领域，开发并提供大量基于互联网的产品与服务，其主要利润来自于AdWords等广告服务。
维基百科

创立于： 1998 年 9 月 4 日，加利福尼亚州门洛帕克
首席执行官： 孙达尔·皮柴 (2015 年 10 月 2 日–)
总部： 加利福尼亚州山景城
总组织： Alphabet (2015–)
创始人： 拉里·佩奇， 谢尔盖·布林
子公司： YouTube， Google.org， Nest Labs， DoubleClick， AdMob，更多

用户还搜索了 还有15+项
亚马逊公司 苹果 雅虎 微软 华为

免责声明 反馈

Figure 1: The result of Google “Google”.

Passing Turing test may only thrill computer scientists, but knowledge based systems actually have some important applications, like search engine. Most widely-used search engines are based on keywords matching. This is a compromising situation because no one would use search engined if he can ask Doctor-who-knows-everything questions directly.

This is not totally a fairy tale, for example, Google added knowledge graph based technologies into its products since 2012 [3]. As shown in Figure 1, the card on the right side shows related information in a structural way. In some way, we consider the machine has knowledge about “Google”, instead of just presenting websites that contains this word in some order.

Knowledge graph is no more than an auxiliary tool like the previous example. However, human will interact with machine more conveniently in the future, directly using natural language. There are several prototype systems already, such as gAnswer [8], Paralex [2], and so on. You can ask questions in natural language on their demo websites.

To build such a question answering system, we first need a knowledge graph. What kind of data can represent knowledge? A tripe: subject, predicate and object. A Subject is usually an entity in real world. A predicate indicates a feature of the subject, with the object as its value. For example, *Abraham Lincoln was born on Feb 12, 1809.* To store such knowledge in computer, we need a structural representation. Two nodes with a directed

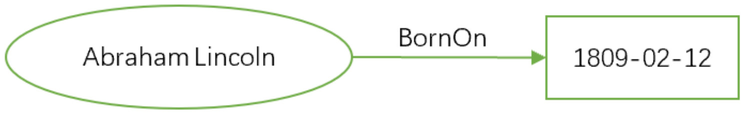


Figure 2: Graph representation of a triple.

edge can represent the previous sentence, see Figure 2. This is so-called resource description framework (RDF) representation. A set of RDF triples then constructs a knowledge graph.

Based on a knowledge graph, we can build a question answering system. There are two procedures in question answering: (1) Natural language questions need transferring to some kind of standard format, with the same semantics. (2) Search in the knowledge graph using the standard format and return the answer. Approaches can be categorized into three classes: **semantic parsing** (transferring questions to query commands based on semantic rules or deep learning, then executing those queries), **information extraction** (extracting entities from questions, matching with knowledge subgraphs) and **vector embedding** (transferring questions into a vector and calculating similarities to filter the answer). In following sections, we will introduce several representative techniques to implement a question answering system.

2 Format Transferring

In 2005, Zettlemoyer and Collins proposed a semantic parsing method using probabilistic combinatory categorial grammar (PCCG) [7]. CCG cam parse a sentence to one or more "logical forms" (lambda-calculus). Essentially, CCG is a lexicon, where every entry consists of a word and its syntactic and semantic type. For example,

$$\text{Utah} := NP : utah$$

is a CCG entry in a geometry knowledge base. NP stands for noun-phrase and is a basic syntactic type. *utah* is a string stands for the semantic value of word "Utah". There are also more complex syntactic types like S/NP, which means this phrase followed by a noun-phrase can construct a sentence. Probabilistic CCG is an extension of CCG. The procedure of CCG parsing can be described as (L, T) pairs, where L is the final logical form and T is the corresponding parsing tree. PCCG defines a conditional distribution $P(L, T|S)$ for a given sentence S. Maximizing this probability solves the ambiguity problem. Their learning algorithm takes a set of sentences and corresponding logical forms, and a incomplete lexicon as input. New lexicon entries are generated from (sentence, logical form) pairs. The weight of each entry in lexicon are adjusted using stochastic gradient descent. Eventually, the algorithm returns a full lexicon and a vector of lexicon-entry weights, which are used in a CCG parser. A new question can then be parsed into lambda-calculus.

Yahya et al.'s approach transfers a natural language question into a SPARQL query [6]. Several third-party tools are used in this approach. They first detect concept and relation phrases from the question. Combining with existing knowledge graph, phrases are mapping to semantic items (candidate spaces), and possible triples are also generated. After those inputs are prepared, they are encoded into an integer linear program and can be easily solved by an ILP solver, which mainly deal with the ambiguity problem. After disambiguation, corresponding triples also come out. Standard SPARQL queries can be generated from those triples.

Unger et al. also transfer questions into SPARQL queries [4]. They first parse a natural language question with a domain tag. Based on its syntactic structure, a SPARQL template is generated. Then they use knowledge information as well as the question to fill this template, forming a SPARQL query.

Zou et al. proposed a work based on information extraction in 2014 [8]. To extract information, they first build a paraphrase dictionary offline. Based on a set of relation phrases and corresponding entity pair sets, they use a graph mining algorithm to find paths in knowledge graph that represent those relation phrases. With a new online natural language question, they build a parse tree and embedding relation phrases into it. Then, a query graph can be constructed based on relation phrases, corresponding entities and values, and the paraphrase dictionary built offline. This query graph is the structured standard format and is used in next procedure to search for the answer. The ambiguity problem is pushed down to next procedure to be dealt with.

Vector embedding is another method. For example, Bordes et al. tried to use the similarity (dot product) between question vector and knowledge subgraph vector to find the answer [1]. In their method, words/entities are encoded in one-hot vectors, thus questions and subgraphs can be encoded in vectors based on their components. They use a matrix M to transfer those vectors to embedding vectors. For example, a question may contain several entities, and we can compare the embedding vector of this question to subgraphs with the same entities to find the answer. Given a set of questions and corresponding answers, we can then training matrix M to get the optimal mapping from an encoding vector to an embedding vector. For a new question, it can also be encoded and embedded to transfer to "standard format", a embedding vector.

3 Answer Searching

This procedure is rather intuitive based on previous section. For those semantic parsing based approaches [7] [6] [4], either lambda-calculus or SPARQL queries, can be directly applied to a RDF knowledge graph to get an answer with mature methods.

Zou et al.'s work [8] are based on information extraction. The previous procedure returns a query graph. They map vertices to entities and relations to predicate paths, and try to find subgraph isomorphism between query graph and RDF knowledge graph. Semantics ambiguity of a single phrase are solved now because subgraph matching of incorrect semantic mappings are unlikely to be found. This knowledge subgraph usually indicates the answer.

As for the last approach [1], on the one hand, an embedding vector for the question is generated. On the other hand, entities appear in the question can also be found in RDF knowledge graph. Thus several candidate subgraphs contains those entities are embedded into vectors to calculate similarities with question vector. With an optimal embedding matrix, the subgraph contains the answer will have the highest similarity value.

4 Summary

In this article, we introduced several basic concepts about knowledge graph and KG based question answering system. There are two phases to solve a natural language question answering problem: transferring the question to some standard format and using it to search for the answer. Three mainstream approaches are based on semantic parsing, information extraction and vector embedding, respectively. We introduced some representative works of those approaches. There are also other works involved with deep learning techniques, which we didn't present in this article. As a conclusion, question answering system is a interesting topic, and also, with great application outlook.

References

[1] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.

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

[3] Singhal, Amit. Introducing the knowledge graph: Things, not strings, 2012. [Online; accessed 7-August-2019].

[4] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM, 2012.

[5] Wikipedia contributors. Turing test — Wikipedia, the free encyclopedia, 2019. [Online; accessed 7-August-2019].

[6] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. Natural language questions for the web of data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 379–390. Association for Computational Linguistics, 2012.

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

[8] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over rdf: a graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 313–324. ACM, 2014.