Department of Computer Science,
University of Hong Kong

Project Plan

Question Answering on Canonicalized Knowledge Base

Author:
Cheng Chen (Caspar)
Xiyang Feng (Andy)
Qianli Song (Charlie)

Supervisor:
Prof. Ben Kao

September 29, 2018

Contents

1 Introduction

Knowledge base question answering (KB-QA) has been a topic of much interest and there have been a lot of studies in both academic and industrial fields. The goal of KB-QA is to construct a question-answering (QA) system that parses a question proposed by the user and leverages its supporting knowledge base (KB) to answer the question. The performance of a QA system is therefore heavily reliant on the correctness, completeness and structuredness of its underlying KB.

A knowledge base can be categorized as either a curated KB or an open KB by the way it is constructed (refer to figure 1). Curated KBs are commonly abstracted as semantic networks composed manually by their community members, while open KBs store semantic assertions, often triples in the form of $\langle subject, relation, object \rangle$, directly derived from unstructured online resources using open information extraction (OIE) techniques [11]. In general, curated KBs are more favored by researchers for building KB-QA systems by their virtue of high-precision and low-redundancy knowledge. Most KB-QA systems nowadays are based on curated KBs, such as Freebase and DBpedia. Nevertheless, given the sheer volume of collective human knowledge, manual maintenance of massive curated KBs can be prohibitively expensive and difficult to scale. QA systems built on curated KBs inevitably suffer from low recall due to their innate incompleteness of factual knowledge [5].

Some researchers have turned to open KB as a more comprehensive source of knowledge, but the performance (accuracy, efficiency and robustness) of open KB on question answering is significantly limited by a common problem called *entity ambiguity*. Entity ambiguity refers to the problem of linking and grouping different surface manifestations (names) of the same real world entity, as well as identifying one among many entities that a certain surface form (name) may refer to [10]. As it is common for an entity to have several different names, and a name to refer to several different entities, open KB contains a tremendous amount of noisy, inconsistent and redundant information. As a result, it can be extremely difficult to match the correct assertion in an open KB without performing thorough entity disambiguation. This process of disambiguating named entities is called *entity resolution*, which is essential for an open KB to be applicable for QA.

Entity resolution can make an open KB more precise, structured and complete, and thus more capable for QA tasks. This project investigates how entity resolution can be done by *canonicalization* (i.e., mapping each name into a canonical form) and how canonicalization enables more effective question answering through redundancy detection, KB compression and knowledge population. In this project plan, we will present our agenda to build a new QA system named *COQA*, short for canonicalized open question answering, which is a QA system built upon canonicalized open KB.

2 Related Studies

2.1 Early QA systems

Early work in Open QA used search engines such as Google Search as the main information source. For example the AskMSR system [1] developed in 2002, mainly relies on the page ranking algorithms of search engines. As AskMSR only uses the summaries of the first three searching results as the knowledge source, there is a high probability for this pool of knowledge to miss the answer to user question. Indeed, the response rate of AskMSR system is only about 50%, and overall accuracy is only 34.7%.

Recent researches have been focusing on large, multi-domain KBs like *YAGO2* [2] and *Freebase* [3], which are mainly curated KBs. These KBs are attractive for QA because they contain highly precise knowledge with little redundancy, which enables

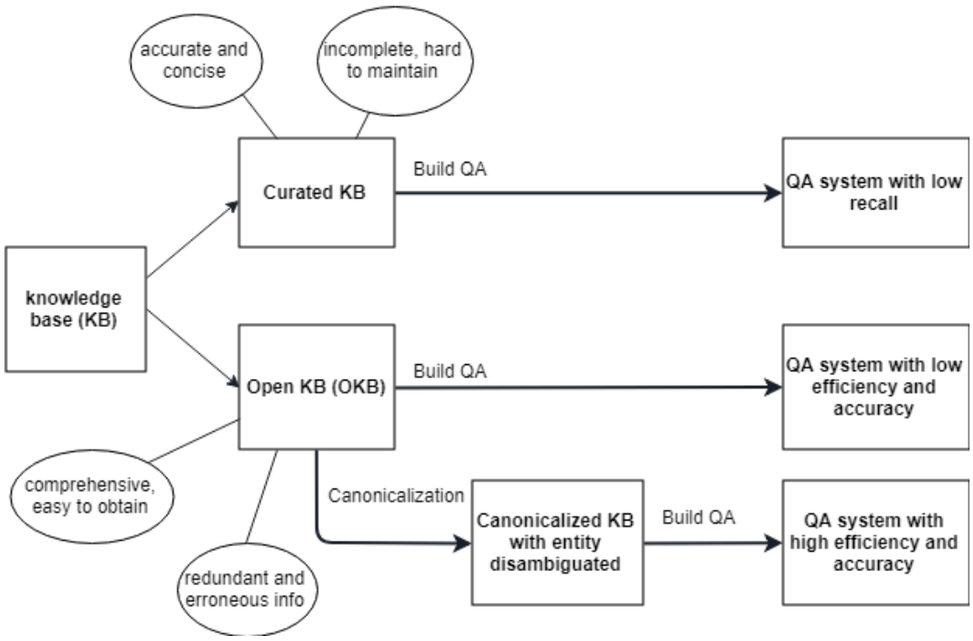

Figure 1: Overview of the background knowledge about QA systems. Our objective is to build a QA system on canonicalized KB, which hopefully has a high recall and precision

QAs to answer questions accurately. However, these systems generally have limited recall, due to the incompleteness of curated KBs. The Paralex system [4] is the first Open QA system to operate over OKB. It uses some learned templates that directly map questions to queries. As one may imagine, this simple method of mapping may result in queries that differ in meaning from the original question, or even don't make sense.

2.2 Recent open QA system: OQA

Later a QA system, named OQA, which can be applied to both curated and open KBs, was developed by the same group of people [5]. This system utilizes a set of KBs, which enable it to combine knowledge extracted from different KBs. This technique also allows it to join multiple assertions to arrive at a single answer. Moreover, OQA also combines high-recall data mining techniques with high-precision, hand-written rules to obtain a query which more precisely represents the original question. However, it still have several problems. Firstly, it fails to fully utilize the contextual information to match the entities. Instead, matching is mainly done by approximate string matching, which is a low-accuracy method. Secondly, the machine learning model is not well trained. The ensemble method performs even worse than individual ones, which should not happen if the component estimators are cleverly combined. As a result, the precision and recall are still not high enough for it to be applicable.

2.3 Current work of canonicalization

Currently no QA system is built on canonicalized KBs. There have been some works which focus on canonicalizing OKBs, and they mainly use the *hierarchical agglomerative clustering (HAC)* method to form clusters of assertions, based on some similarity functions to determine whether different assertions should be mapped to the same cluster [8]. However, this process is extremely time-consuming, as the similarity measure is computed between massive pairs of assertions, and the iterative process never ends until a certain threshold of dissimilarity between clusters is reached. As the similarity function is also computed based on common substrings between different entities, it is

also not accurate enough. With the low accuracy and efficiency in place, the current canonicalization methods cannot be directly used in building QA systems.

3 Problems to be solved

The ultimate goal of our project is to improve the performance of QA systems using canonicalized KB. With regard to the major steps involved in building COQA presented in Figure 2, our team aspires to overcome the following five hurdles.

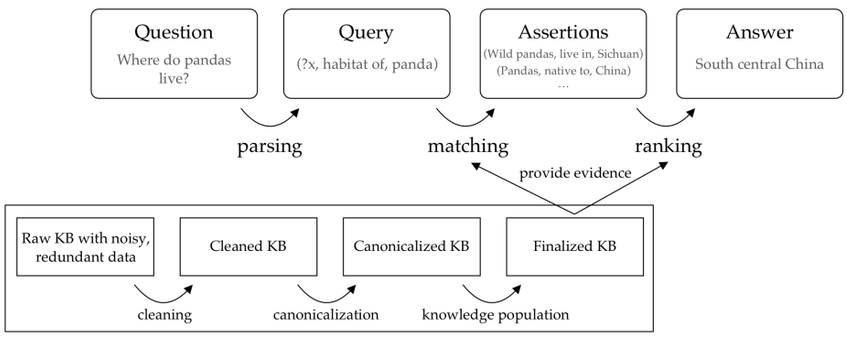

Figure 2: Breakdown of all major steps influencing COQA's performance

3.1 Accuracy and efficiency of entity clustering

Entity clustering, which is the first step of canonicalization, requires a pairwise function that quantifies the similarity between two candidate entities. Intuitively, a strong similarity function should be able to reflect both string similarity and contextual similarity. The *state-of-the-art* similarity function used currently, however, is purely based on word frequency and string matching. Given the ubiquity of synonyms with dissimilar spellings (e.g. 'New York City' and 'Big Apple', 'Xiamen' and 'Amoy'), it is reasonable to believe that similarity functions with more sophisticated context-reading capabilities could effectively boost the accuracy of entity clustering. Efficiency is another major concern, as the clustering process has been proved to be extremely time-consuming.

3.2 Information loss due to canonicalization

Given a cluster of synonym phrases, the *state-of-the-art* canonicalization process proposed selecting a representative phrase to replace the other phrases in the same cluster, or instead making use of certain coding scheme to give every entity a unique code. This works for the sole purpose of deduplication or KB compression, but it will potentially cause substantial damage to the completeness of the KB and therefore compromising the accuracy of the supported QA system. The canonicalization process has to be reinvented with the question-answering objectives in mind.

3.3 Ranking models on canonicalized KB

The ranking models used in current QA systems for finding matching entities in a KB are believed to be poorly trained. Our team will propose new machine learned models and train them well for them to be effectively adapted to canonicalized KB. To accomplish this, we need to first design the components of the feature vector that can effectively distinguish between different entities. With the feature vector in hand, our well-trained model can be applied to it and give a representative score for every entity mention.

3.4 Knowledge population

By clustering synonymous entities, canonicalization detects new relationships that can be derived from the combination of several existing assertions. Consider the instance where the KB contains ⟨"Mary","marries","John⟩ and ⟨"Marry","is the mother of", "Henry"⟩. If the user asks "who is the father of Henry", the current QA systems will not be able to answer it. If this can be solved, the QA system will be able to answer much more complex questions.

3.5 Improvement of parsing techniques

The current parsing technique also needs to be improved, as sometimes the user questions may be misrepresented by the parsed query. A major problem lies in the parsing of constraints. For example, if the user asks "who is the former president of the US", the parsing program may not be able to recognize the word "former" as a significant constraint, therefore the QA system may ultimately return "Donald Trump" instead of "Barack Obama" as the result. However, this will not be our main focus, as the current parsing techniques are already doing quite well, and the problems stated above do not always occur.

4 Methodologies

4.1 Process of building COQA

We will follow the following steps to build our QA system (refer to figure 2):

- Find a medium-size open KB that is suitable for a final year project. Around 1 million assertions will be the right size.
- Adapt the current canonicalization code to build a *canonicalized KB*, which is written mainly in Java. The main area that we are aspired to improve is the similarity function. We will try to plug various contextual factors into the similarity function, implementing a number of approximation methods, in hope of making the clustering process accurate as well as efficient.
- Build a *QA system* on top of the canonicalized KB. The current QA systems need to be adjusted for them to be applied on canonicalized KB, which has totally different feature vectors. Various machine learning methods and fine-tuning will also be investigated, in order to improve its probability to select out the most suitable matching assertion. This part will probably be written in Python, as this part is not particularly time-consuming compared to the canonicalization process, so a more high-level language can be utilized.
- Implement an *automatic populater* which can populate the KB automatically, based on the linking of current assertions as well as natural language processing to deal with sentence semantics.
- Investigate different *parsing methods* to improve the probability of right parsing of the proposed question.

4.2 Evaluation metrics

As has been stated above, applying canonicalization is mainly for improving the performance of QA system. To see the degree to which the performance has been improved, we will need some evaluation metrics to compare the performance of *OQA* and our product *COQA*. We will mainly use the following metrics:

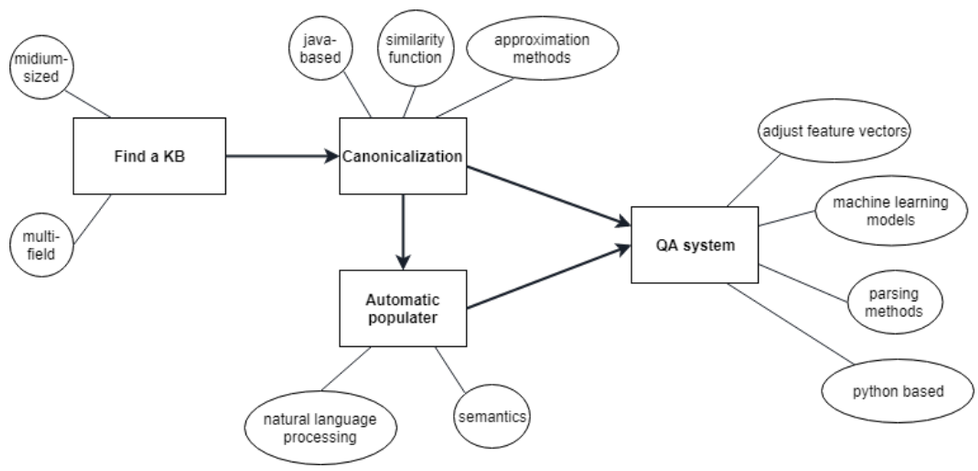

Figure 3: Approach by which we will develop our product. The whole process consists of 4 main phases, represented by rectangles, and the main features are noted in ellipses.

- **Recall** reflects the overall ability of our QA system to answer various questions across different domains. It is defined as:

$$recall = \frac{|\{\text{questions answered correctly}\}|}{|\{\text{questions proposed}\}|}$$

  A high recall requires the breadth of knowledge of the underlying KB, and therefore this is where existing curated KB-QA systems consistently underperformed. It also requires our COQA system to parse questions correctly to form the representative queries.

- **Precision** factors in all questions answered by the QA system and determines how precise these answers are. It is defined as:

$$precision = \frac{|\{\text{questions answered correctly}\}|}{|\{\text{answers returned}\}|}$$

  Precision is the main weakness of an uncanonicalized open KB compared with curated KBs in question-answering tasks, due to the large amount of noisy data a raw open KB normally contains. In this project, precision will reflect the effectiveness of our canonicalization approach. A high precision is also highly dependent on correct training of the machine learning model to rank results.

- **F1 Score** is the harmonic average of precision and recall, ranging from 0 to 1. That is:

$$F1 = \frac{2 \cdot recall \cdot precision}{precision + recall}$$

  F1 score seeks a balance between recall and precision. Essentially, one of the goals we set out to achieve with a canonicalized open KB (instead of a curated KB) is to balance *more* data (completeness, favoring recall) with *better* data (correctness, favoring precision). Theoretically, canonicalization is expected to improve both recall and precision up to a certain level. Beyond that level, a higher recall may come at the expense of a lower precision, and vice versa. This recall/precision trade-off will be carefully studied in order to reach a reasonable compromise between the two.

Our team will first focus on achieving a high recall, to make sure our system can answer a grand scope of questions. We will then try to improve the precision by improving every step from question parsing to result ranking. The combined improvement of recall and precision is believed to give us a huge improvement in terms of F1 score.

5 Feasibility analysis

Despite all the progress already made in KB-QA and canonicalization, there are still a few challenging problems yet to be properly solved, many of which have been discussed in Section 3. Question-answering on canonicalized KB is also a novel topic with no previous realizations found, so the project is not free from theoretical and practical challenges. The main challenges are summarized as follows.

5.1 Risks

5.1.1 Time taken to process large KBs

From entity clustering to training ranking models, a large knowledge base with millions of assertions poses severe challenges to run time and may render theoretically-effective algorithms impractical to use. Luckily, an algorithm proposed recently has effectively carried out orders-of-magnitude speedups for entity clustering, shortening the canonicalization process from months to minutes [12]. In order to practically handle massive open KBs, our team will need to devise similar efficient techniques to speed up any algorithm concerning the processing of large KBs.

5.1.2 Ground truth for training ranking model

To train the ranking model used for selecting the answer from candidate assertions, we need a collection of question-answer pairs as the labeled training data. Since such datasets are not readily available for open KBs, we may need to tackle the problem in the following two ways. The first is to transform and leverage on question-answer-pair datasets that are based on curated KBs. SimpleQuestions and WebQuestions [5] are two commonly used KB-QA benchmarks. The second is to apply learning methods on open KBs to generate natural language questions using facts, so that we can produce large-scale question-answer corpora for a specific KB. The documentation of OQA systems can be referred to for training the individual component models.

5.1.3 Implementation of automatic populater

An automatic populater can link assertions together and consequently discover new relations. A successfully canonicalized KB will automatically group the same entity in the same cluster. These clustered identities may be further related and form new relations. On top of that, we hope to apply natural language processing on the clustered assertions, understand their semantics and derive new relationships from them. There are several NLP tools such as *Stanford CoreNLP* and *AllenNLP* that we can make use of to perform logical reasoning [7].

5.1.4 Accuracy-efficiency tradeoff

Canonicalization will make question-answering more efficient by compressing the knowledge base. But as demonstrated in Section 3, while deduplicating redundant data, canonicalization may take away valuable information and hurt the accuracy of question-answering tasks. Therefore accuracy and efficiency need to be well balanced in order to achieve a satisfactory overall performance.

5.2 Limitations

Given the inexhaustible variety in natural language and knowledge representations, robustness is an inherently difficult challenge, as a slight tweak in wording or query formulation may easily confuse the system. Many researchers have tried to develop techniques such as question decomposition and rule mining to achieve a higher robustness. In this project, a high robustness will not be our primary goal until we approach the final stage of our development.

5.3 Overall assessment

Since both KB-QA and canonicalization are well-researched areas, there are quite a few existing projects with comprehensive implementation details to refer to. These projects will significantly facilitate our setup of preliminary architectures as well as performance benchmarks for subsequent implementations. Hence, building a working prototype for COQA is not surrounded by much difficulty.

In theory, caconicalization alone, if implemented correctly, could give a huge boost to the efficiency (and accuracy) of question-answering. Our ability to solve most of the theoretical and practical problems mentioned above leads to extra improvements in performance that our product can achieve. As demonstrated above, these challenges are mutually independent components and each could be readily replaced with a "baseline approach". In addition, these challenges are highly addressable by the methodologies mentioned above, which guarantees a considerable level of feasibility for our product.

6 Proposed Schedule

Generally speaking, each person in our group will be in charge of one aspect of the project, which is specified in the following way:

- Song Qianli: canonicalization of KB
- Chen Cheng: implementation of automatic populater
- Feng Xiyang: implementation of QA system

The detailed schedule is as follows:

Deadline	Deliverables
30 September 2018	Detailed project plan Project web page
30 October 2018	Reimplementation of the current canonicalizer
20 December 2018	Improvement of the performance of canonicalizer Implementation of OQA framework
7-11 January 2019	First presentation
20 January 2019	Preliminary implementation Detailed interim report
28 February 2019	Implementation of an automatic population system QA system integrated with canonicalized KB
14 April 2019	Finalized tested implementation Final report
15-19 April 2019	Final presentation
29 April 2019	Project exhibition
29 April 2019	Project competition

7 Conclusion

Currently all the question answering systems based on open KB suffer a lot from low accuracy and efficiency, consequently they are typically considered not applicable. This is mainly caused by the significant problem of entity ambiguity, which makes assertion matching difficult. To address this problem, we propose to apply canonicalization to KBs in order to make assertion matching much easier. We will implement a QA system on canonicalized KB, namely COQA, which hopefully has a satisfactorily high recall

and precision. After thorough analysis, our project is believed to be highly feasible as well as innovative. In the remaining 7 months, we will stick to our schedule, trying to reuse the current available implementations as much as possible, while investigating on adapting and improving them via various methods. The progress will be updated on our website from time to time. If the final implementation is successful, we will publicize our product and make it available for other people to make use of.

References

[1] M. Banko, E. Brill, S. Dumais, and J. Lin. AskMSR: Question answering using the worldwide web. In 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases, 2002.

[2] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. Natural Language Questions for the Web of Data. In EMNLP, 2012.

[3] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In EMNLP, 2013.

[4] A. Fader, L. Zettlemoyer, and O. Etzioni. Paraphrase-Driven Learning for Open Question Answering. In ACL, 2013.

[5] Fader, A., Zettlemoyer, L., & Etzioni, O. Open question answering over curated and extracted knowledge bases. Paper presented at the Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.

[6] Vashishth, S., Jain, P., Talukdar, P. CESI: Canonicalizing Open Knowledge Bases using Embeddings and Side Information. Paper presented at the Proceedings of the 2018 World Wide Web Conference, Lyon, France. 2018.

[7] Galárraga, L. A., Teflioudi, C., Hose, K., Suchanek, F. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. Paper presented at the Proceedings of the 22nd international conference on World Wide Web. 2013

[8] Galárraga, L., Heitz, G., Murphy, K., Suchanek, F. M. Canonicalizing open knowledge bases. Paper presented at the Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. 2014.

[9] Shen, W., Wang, J., Han, J. Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering, 27(2), 443-460. 2015

[10] Ratinov, L., Roth, D., Downey, D., Anderson, M. Local and global algorithms for disambiguation to wikipedia. Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. 2011.

[11] Yin, P., Duan, N., Kao, B., Bao, J., Zhou, M. Answering questions with complex semantic constraints on open knowledge bases. Paper presented at the Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 2015.

[12] Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, Pengcheng Yin. Towards Practical Open Knowledge Base Canonicalization. 2018.