Probabilistic Knowledge Base assisted Question Answering

Christan Grant, Kun Li, Yan Chen, Daisy Zhe Wang
University of Florida
Computer & Information Science & Engineering Department
Gainesville, Florida
{cgrant,kli,yang,daisyw} @ cise.ufl.edu

ABSTRACT

Abstract

1. INTRODUCTION

Motivate Prob KBs
Describe Motivation of the the KHop system.
Give example scenario.
Introduce the Khop system.
Describe intro that this demo show real time incremental changes and probability changes to a large KB.

2. SYSTEM OVERVIEW

This demonstration describes a question answering systems that leverages a probabilistic knowledge base. To introduce this section, we first give a walkthrough of how a question is evaluated. We then give a detailed description of each component involved in question answering computation.

A user user first enters a question q. We then leverage the SEMPRE system to translate the question into SPARQL, the de facto language for querying RDF data stores. SPARQL is an SQL-like language that allows declarative subgraph expressions. We can then take the SPARQL query $s(q)$ and extract the supporting triples $t_{s(q)}$ that underly the subgraph. These triples are in the form $\langle s, p, o\rangle$ correspond to the facts that support the answer to the question. To obtain the $t_{s(q)}$, we fan evaluate the SPARQL query and materialize all the intermediate triples. We can use $t_{s(q)}$ to search our probabilistic knowledge base of facts to obtain the closest matching facts. Each fact $f \in \mathcal{F}$ is of the form $R(A, B)$ and a triple $t_{s(q)}$is equivalent to a fact f when $(A \simeq s)\&(R \simeq p)\&(B \simeq o)$. Given the equivalent facts, we use our k-hop algorithm to determine the joint probability that each fact is correct. This probability is a truthfulness score that can be paired with each answer to the question.

In the next few sub-sections we describe each part of the process. We start with the interface where the user can enter

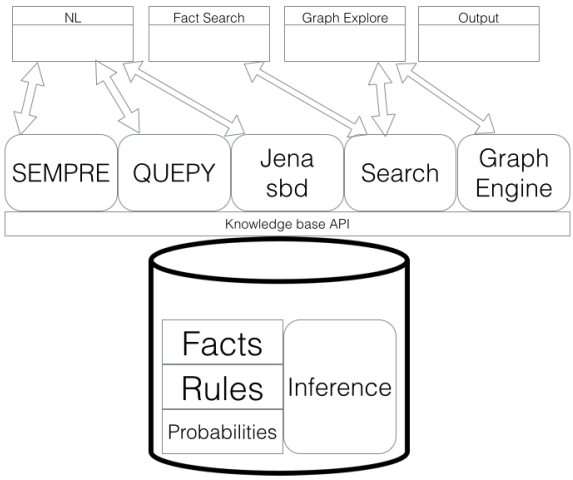

Figure 1: Question Answering system architecture.

natural language questions and a description of how SEMPRE processes the question (Section 2.1). Next, we describe how we perform lookups for candidate facts (Section 2.2). We then define our method to derive the joint probability distribution (Section 2.3).

2.1 Interface

The interface allows users to make queries using three different modalities. Users will be able to enter natural language questions, search through the set of existing facts, and use a graph to explore connections between graphs. New probabilistic facts and rules can also be added to the system through the interface. Users can also remove or alter the existing facts and rerun queries. The status of queries and the underlying processes are displayed on the main interface.

When a user enters a natural language utterance q, we use the SEMPRE 2.0 system to transform the utterance to a logical form [1, 2]. We then translate the logical form to SPARQL for execution over the knowledge base $\mathcal{D}$. Let $s(q)$ be a function that transforms a natural language utterance to the SPARQL query. We then parse the SPARQL query and extract the intermediate triples $t_{s(q)} = \{\langle s, p, o\rangle \ldots\}$. Like the SPARQL query, these triples only specify a template of the facts that are required to evaluate the utterance. We evaluate the SPARQL query over $\mathcal{D}$ to obtain an answer

α, we map the query template to define the candidate set of triples $t^{\alpha}_{s(q)}$.

For each triples in $t^{\alpha}_{s(q)}$ we perform a look up in $\mathcal{D}$. If there is an exact match, the triple is mapped to a score of 1. If there is no exact match in $\mathcal{D}$, we estimate the probability of the triple appearing using a straight-forward application If found, we assign the triple to a value that corresponds to the product of a normalized IDF-like weight for each partial match. The intuition behind this weighting is that if a fact does not exist we would like to compute the weight that corresponds to the possibility that the facts could exist. An equation representing this value is as follows,

$$\omega(s,p,o) = \begin{cases} 1 & \text{if } exists(\langle s,p,o \rangle) \\ \max(\omega(o,s,p), P(s,p,o)) & \text{otherwise,} \end{cases}$$

where $P(s,p,o) = P(s|p,o) * P(p|o) * P(o)$.

We then use the k-hop algorithm to estimate the joint probability of a fact existing giving the rules.

Describe the purpose translation of natural language questions queries. Add the auto complete for previous questions.

Listing 1: Algorithm for obtaining the information

```
def answer(q):
  t = {'q': q,
    's': SEMPRE.toLogicalForm(q)}
  t['a'] = SEMPRE.toSPARQL(t['s'])
  # Do fact search
  # Evaluate query
  # Link probabilities
  return t
```

2.2 Fact Search

Describe how facts are searched using the database. Describe how results are ranked. Describe how new results are discovered.

Describe D3 visualization of graph and rule display Describe user interaction with graph Describe user selecting facts Describe users removing facts

2.3 Logic

Describe the translation of NL-to-queries using templates (quepy) and also sempre.

Describe how rankings are computed from queries. SEMPRE Returns probabilities, Quepy gives a 0 or 1 probability. We compare the fact probabilities to the sempre results.

2.3.1 Knowledge Base

Desscribe the PostgreSQL database and the other serices running on servers. Describe the tables Describe the functions that are called Describe the parallelism

The system is loaded with docker, a system container, so any modifications by demo can be quickly rolled back to the intial state.

3. RELATED WORK

Describe incremental KB from Chris Re

Question answering over freebase [3].

Describe Google Knowledge Vault Describe Fact finding in Google KB Describe Source finding.

4. DEMONSTRATION

Describe the demo setup.

Describe how users will be able to alter parameters.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[2] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544, 2013.

[3] X. Yao and B. Van Durme. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*, 2014.