Knowledge-Based Question Answering

Sudipto Ghosh

M.Sc. CS Semester I
Department of Computer Science
University of Delhi

January 20, 2023

Overview

1. Problem Statement
   - Problem Definition
   - Motivation
   - Modelling
2. Modules
   - KB Construction
   - Question Understanding
   - Inferencing Engine
3. System Overview
   - Use Case Diagram
   - System Architecture

Problem Definition

- Parse unstructured text from domain corpus, identify entities, extract relations, map relations to domain concepts and build knowledge base.
  Input: Corpus, domain ontology, and training examples consisting of entity boundaries, relationship dependencies, valid triples.
  Output: $<$s,p,o$>$ triples to populate the knowledge graph.
- Model natural language question into a query, infer the facts about it required for the answer, assemble the facts into a natural language answer, and present it to the user.
  Input: Question as a spoken utterance or text prompt.
  Output: Answer/fact as a spoken utterance or text response.
- Ensure system maximizes performance on giving correct answers to a set of competency questions.

Motivation

- Leveraging domain knowledge to improve virtual assistants.
- Inferencing step $\implies$ system can answer unanticipated questions.
- Level of detail can be controlled to suit the expertise of the user.
- Can we answer complex questions that contain multiple subjects, express compound relations, or require simulated thinking?

Example

Q: What is the capital of India?
A: The capital of India is New Delhi.
Q: What is the state of motor 2?
A: Motor 2 is currently turned off.

Modelling

Information Retrieval Agent

Given documents forming the domain corpus D, extract facts and construct a knowledge graph containing entities and semantic relations between them. Percepts are continuously made available in light of quick-updating corpus, knowledge in the KG may be revised based on incoming facts.

Conversational KBQA Agent

Given a KG built from facts in domain corpus D and a conversation history $(Q_1, A_1, Q_2, A_2, \ldots)$, the goal for an reading comprehension / knowledge based conversational QA agent is to answer the n^{th} question Q_n using the whole conversation history as context.
However, in an open domain conversational QA, user can switch between topics, making the conversation history irrelevant.

KB Construction

- Knowledge base construction (KBC) is the process of populating a knowledge base (KB) with facts (or assertions) extracted from data.

Sentence

Paracetamol, also known as acetaminophen, is usually prescribed for treating fever

Entity Recognition

Paracetamol, also known as acetaminophen, is usually prescribed for treating fever

Relation Extraction

Paracetamol, also known as acetaminophen, is usually prescribed for treating fever

Coreference Resolution

Paracetamol, also known as acetaminophen | Paracetamol is usually prescribed for treating fever

KB Construction (contd.)

Triple Extraction

<Paracetamol, known as, acetaminophen>
<Paracetamol, prescribed for, fever>

Entity Linking

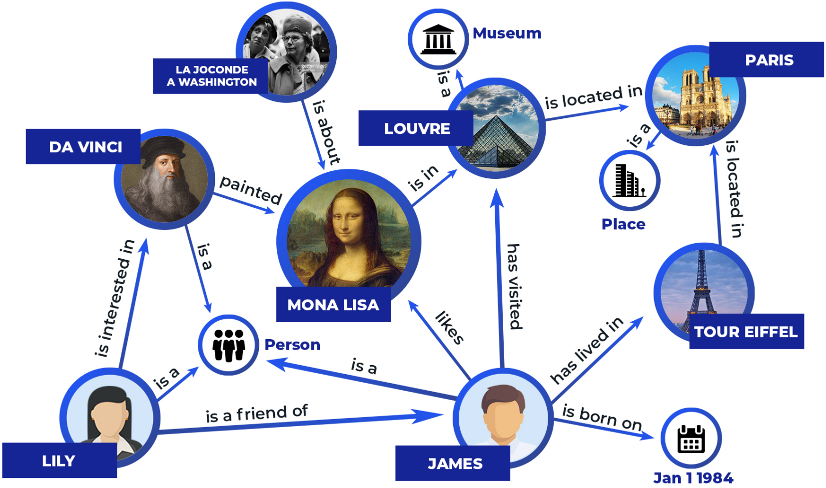
<https://en.wikipedia.org/wiki/Paracetamol, known as, https://en.wikipedia.org/wiki/Acetaminophen>
<https://en.wikipedia.org/wiki/Paracetamol, prescribed for, https://en.wikipedia.org/wiki/Fever>

Ontology Mapping

prefix wikidata https://www.wikidata.org/wiki/Property

<https://en.wikipedia.org/wiki/Paracetamol, wikidata:P2561, https://en.wikipedia.org/wiki/Acetaminophen>
<https://en.wikipedia.org/wiki/Paracetamol, wikidata:P2176, https://en.wikipedia.org/wiki/Fever>

- The resultant triples are stored to a triple or RDF store like Blazegraph through semantic queries, or a graph database like Neo4J.



Source: Let The Machines Learn

How?

- Data Collection - scrape domain repository, select domain ontology.
- Use rule-based approaches (e.g. JAPE Grammar) to gather facts from the documents?
- Use language models (e.g. BERT, RoBERTa) and fine-tune it to do the fact extraction?
- Representation Format - RDF, Turtle, N3 in Triple Store - KG.
- Entity Linker needs a KG/Linked Open Data to map entities to.

Turtle - Terse RDF Triple Language

RDF Graph in a compact, textual form.

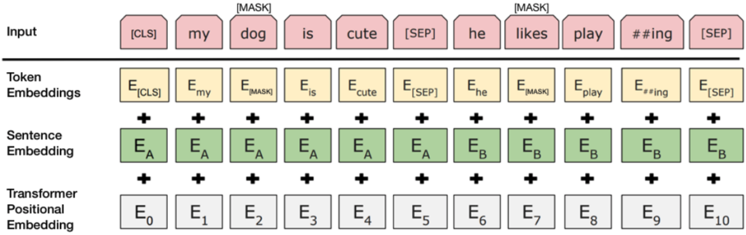
<http://example.org/#spiderman> <http://www.perceive.net/relationship/enemyOf> <http://example.org/#green-goblin> .

<http://example.org/#spiderman> <http://xmlns.com/foaf/0.1/name> "Spiderman" .

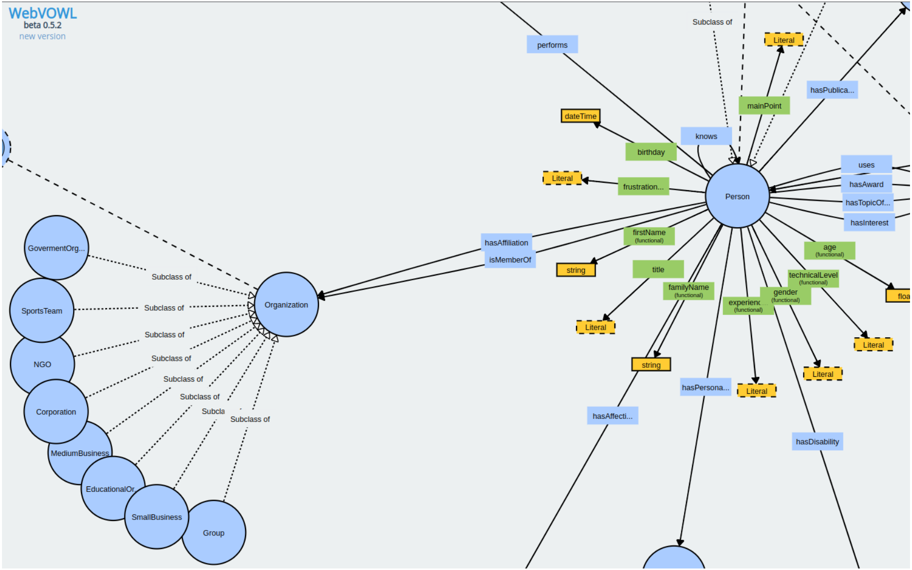
JAPE Grammar

```
Rule: BENCH
Priority: 20
(
  (
    {Token.string == "BENCH"}
  ):Type
  {Token.string == ":"}
  (
    {Token.kind == "word", Token.category != "CC"} |
    {Token.string == ".", Token.category != "CC"}
  )+ :Name
  ({Token.kind == "punctuation"})?
  ((
    {Token.kind == "word", Token.string != "judgment"} |
    {Token.string == ".", Token.sting != "judgment"}
  )+) :Name2
)
–>
:Type.COURT_OFFICIALS = {kind = "BENCH", rule = "BENCH"},
:Name.JUDGE = {rule = "BENCH"},
:Name2.JUDGE = {rule = "BENCH"}
```

BENCH:Justice D. Y. Chandrachud, Justice R. K. Krishna



Source: BERT (Devlin et. al., 2018)



Source: PersonasOnto

Question Understanding

- Understanding natural language questions refers to the ability to break down a question into the requisite steps for computing its answer.
- Encode questions into low-dimensional vectors with contextual information?
- Calculate semantic similarity between questions and entities in KB?
- Detect and link entities in questions to those in KB and construct queries?

Question

What drug is prescribed for treating fever?

Parsed Query

<?, prescribed for, fever>

Semantic Query

<?, https://www.wikidata.org/wiki/Property:P2176, https://en.wikipedia.org/wiki/Fever>

How?

- Use language model to convert natural language questions to queries.
- Use the correct pattern matching strategy - SPARQL.
- Slot-filling values from question to queries.
- Running the query and retrieving triples.

SPARQL - Querying Language

Data:
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
<http://example.org/book/book2> <http://purl.org/dc/elements/1.1/title> "Artificial Intelligence" .

Query:
SELECT ?book ?title WHERE { ?book <http://purl.org/dc/elements/1.1/title> ?title . }

Result:
<http://example.org/book/book1>, "SPARQL Tutorial"
<http://example.org/book/book2>, "Artificial Intelligence"

Question Understanding (contd.)

- Generate semantically similar queries?
- Rank templates matching question?

KB Inferencing

Given a question Q, match that with a template that is semantically similar and find a set of triples from the KG that are similar to the entities or relations in the question, find top K triples that might answer the template.

Example

Question: What are the different book titles that are available?

Query:
SELECT ?book ?title WHERE { ?book <http://purl.org/dc/elements/1.1/title> ?title . } ($\tilde{9}$9%)
SELECT ?book ?title WHERE { ?book <http://random.org/concept/movie-title> ?title . } ($\tilde{1}$%)

Result:
<http://example.org/book/book1>, "SPARQL Tutorial"
<http://example.org/book/book2>, "Artificial Intelligence"

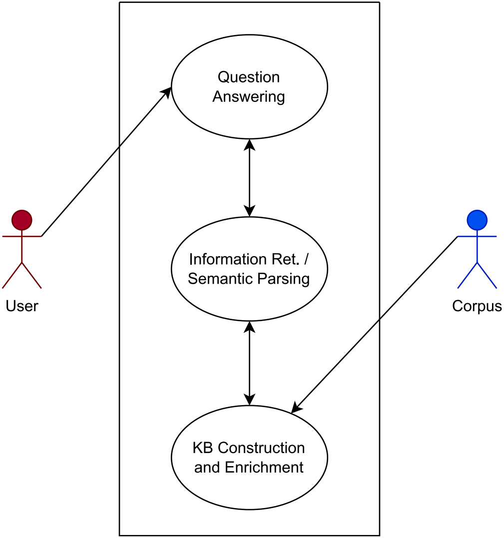
Inferencing Engine

- KBQA Models learn Question Answering by using a QA corpus and a populated KB – uncertainty, incompleteness and noise are inevitable
- Probabilistic Inferencing $\implies$ infer predicates from templates.
- Offline – learn the mapping between templates and predicates.
- Online – break questions down to simple questions, make use of probability distributions, calculate maximum likelihood.
- Entity distribution, template distribution, value (answer) distribution.
- Questions in actual interactions might be vague and unusual.
- Answer questions with entities/predicates matching the top confidence score.

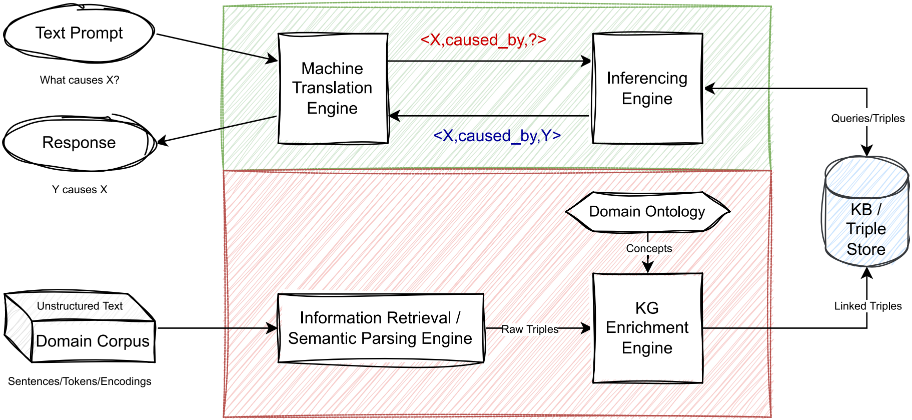
Example

Template: what is treated by $medicine?
Predicate: prescribed_for (maybe 90.5%), founder (maybe 0.01%)

Use Case Diagram



System Architecture



References

[1] Junwei Bao et al. "Knowledge-Based Question Answering as Machine Translation". In: *Proc. of the 52nd AM of the ACL*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 967–976. DOI: 10.3115/v1/P14-1091. URL: https://aclanthology.org/P14-1091.

[2] Vasudha Bhatnagar et al. "Similar Cases Recommendation using Legal Knowledge Graphs". In: *Proc. of KDD 2021*. DOI: 10.48550/ARXIV.2107.04771.

[3] Sudipto Ghosh et al. "Constructing a Knowledge Graph from Indian Legal Domain Corpus". In: *Proc. of TEXT2KG 2022 and MK 2022 co-located with ESWC 2022*, pp. 80–93. URL: http://ceur-ws.org/Vol-3184/TEXT2KG_Paper_6.pdf.

[4] Daniel Jurafsky and James H. Martin. "Question Answering". In: *Speech and Language Processing*. USA: Prentice-Hall, Inc., 2021. Chap. 23. ISBN: 0131873210.

[5] Yunshi Lan et al. "Complex Knowledge Base Question Answering: A Survey". In: *Proc. of IJCAI 2021*. DOI: 10.48550/ARXIV.2108.06688.

[6] Ce Zhang. "DeepDive: A Data Management System for Automatic Knowledge Base Construction". PhD thesis. 2015. URL: https://cs.stanford.edu/people/czhang/zhang.thesis.pdf.

Thank You