2018-06-20

Core techniques of **QA Systems over KBs** a **Survey**

Core techniques of
QA Systems over KBs
a Survey

Guillermo Echegoyen Blanco

Overview

1. Intro
2. Tasks
3. Question Analysis
4. Phrase Mapping
5. Dissambiguation
6. Query Construction
7. Conclusions
8. End
9. References

Intro

- A Question Answering System should be able to: *Understand a Natural Language Question so as to be able to answer based on some pre-known data.*

- Typically involves accepting a question and generating a SparQL query capable of extracting the information which answers the user question.

- QALD benchmark
- WebQuestions benchmark
- SimpleQuestions benchmark

2018-06-20

Core techniques of **QA Systems over KBs a Survey**

└─Tasks

└─Tasks

- Question Analysis
- Phrase Mapping
- Disambiguation
- Query Construction

Question Analysis #1

Analyze syntactic features to extract meaningful information:

- Type of question (is it a Which, What. . . question).
- Multilinguality (is it in English, French. . .).
- Correspondance to KB entities/classes.
- Tokens in the sentence and it's relations.
- Useless words in the sentence.

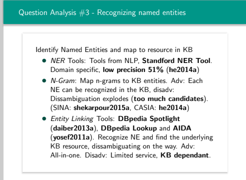
Question Analysis #2

Techniques based on:

- Recognizing Named Entities
- Segmenting with *POS** Tags
- Identifying dependencies using parsers

POS Tag: Part-Of-Speech Tag

- Recognizing Named Entities consists in finding the entities corresponding to parts of the phrase (eg: Europe dbr:European_Union):Which token correspond to which resource in the KB
- Segmenting is like tokenization of different parts of the string, where the tag is usually universal
- Dependencies refer to parts of the phrase which depend upon others, direct cumpliment, adjective, subjective noun. . .



Identify tokens in the sentence that refer to a resource in the KB, discarding useless words.

- When grouping n-grams, if an entity is found, the n-gram is considered, else more n-grams are tried.

Propose n-grams with attention mechanism?

Identify which phrase correspond to instances, properties, classes. . . and which is irrelevant.

- *Handmade rules*: Regular expressions depenending on question type, structure. . . . (PowerAqua **lopez2012a**, Treo **freitas2014a**, DEANNA **yahya2013a**). Disadv: **regex built by hand**.

WRB	VBD	DT	NNP	NNP	VBN	.
When	was	the	European	Union	founded	?

Figure: POS tagging from the **Standford POS Tagger**

- *Learning rules*: **Machine Learning** approach, train over corpus (Xser **xu2014a**, UTQA **pouran2016a**). Disadv: **training corpus needed**.

none	V-B	C-B	none	none	E-B	E-I	R-B	.
By	which	countries	was	the	European	Union	founded	?

Figure: Question annonated with **CoNLL IOB format**

It is not clear how to identify the relation between different chunks of a question

Grammar based parsers to generate trees or DAGs

- *Dependency grammars*: **Standford dependency parser**, word dependencies. Adv: can extract relations along with it's arguments (gAnswer **zou2014a**, **PATTY nakashole2012a**)
- *Dependencies and DAGs*: Dependencies between phrases. Disadv: **parser trained on dataset** (Xser **xu2014a**).

DAG based parser operates on a phrase level, dependency grammars on a word level.

2018-06-20

Which techniques to choose?

- Xser (**trained DAG**) reports best results on *QALD 4.1 & 5*
- gAnswer (**Dependency grammars**) reports fastest results on *QALD 3 & 4*

Machine Learning approach: Can be fast enough and there is plenty of data available.

Phrase Mapping #1

Find the resources in the KB with the highest probability that maps to the phrase.

Problems:

- String similarity
- Semantic similarity
- Language

String similarity: very similar words, different meaning (which, witch) Semantic similarity: words with related semantic meaning but different writing (king, queen)

Phrase Mapping #2

- Database with lexicalization: *WordNet*, *Wiktionary*, *PATTY* Expand the phrase with synonims and use that for search. Adv: High number of candidates, disadv: **Big search space**, **not very useful for domain specific mappings**.
- Mappings using large texts: **word2vec** semantics reflected in the associated vector. Adv: aids in **lexical gap, string similarity and semantic similarity**, disadv: **needs training on large texts, noisy, performance**.

PATTY is a database with relational lexicalization, uses pattern synsets (is album, [[num]] album by)

Phrase Mapping #3 - Summary

Which techniques to choose?

ToDo

Disambiguation #1

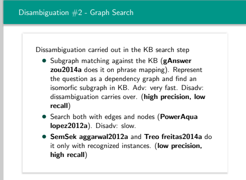
QA systems generate lots of possible interpretations due to language ambiguities and search process.

- Find univocally the resource that maps to the requested question.

Typically approached:

- String or semantic similarity to resource label (include).
- Consistency check between the properties and their arguments (exclude).

Local dissambiguation excluded, all systems do it. Example "Who is the director of The Lord Of the Rings?", with no information associated with the director resource, it is not possible.


Disambiguation #2 - Graph Search

Dissambiguation carried out in the KB search step

- Subgraph matching against the KB (**gAnswer zou2014a** does it on phrase mapping). Represent the question as a dependency graph and find an isomorfic subgraph in KB. Adv: very fast. Disadv: dissambiguation carries over. (**high precision, low recall**)
- Search both with edges and nodes (**PowerAqua lopez2012a**). Disadv: slow.
- **SemSek aggarwal2012a** and **Treo freitas2014a** do it only with recognized instances. (**low precision, high recall**)

gAnswer uses scores each possible match proportionally to the distance between labels and resources, searches both in edges and nodes.

PowerAqua does a balance between recall and precision based on the question analysis.

2018-06-20

Core techniques of **QA Systems over KBs a Survey**

└─Dissambiguation

└─Dissambiguation #3 - Graph Search

Dissambiguation #3 - Graph Search

gAnswer searches in the edges and vertices, **SemSek, Treo** search on instances and properties attached.

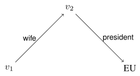
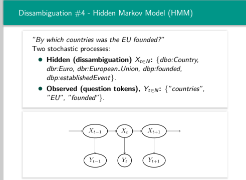


Figure: Subgraph generated for the question "Who is the wife of the president of the EU?"



This means that the appearance of a resource at time t depends only on the appearance of a resource at t -1

Core techniques of **QA Systems over KBs a Survey**
└─Dissambiguation
└─Dissambiguation #5 - HMM

The problem is reduced to find the most probable set of states. Extra parameters:

- Initial probability $P(X_0 = x)$ for $x \in X$
- Transition probability $P(X_t = x_1 | X_{t-1} = x_2)$ for $x_1, x_2 \in X$
- Emission probability $P(Y_t = y | X_t = x)$ for $x \in X, y \in Y$

It is not necessary to know the the dependency between different resources, just the available resources.

Disambiguation #6 - HMM

SINA (shekarpour2015a): slow

- Emission: string similarity between label and segment.
- Initial & Transition: estimated based on the distance of the resource in the KB and popularity.

RTV (giannone2013a): inaccurate

- Emission: word embeddings
- Initial & Transition: uniform across all resources

Dissambiguation #7 - ILP & MLN

ILP Optimization problem

- **DEANNA** (**yahya2013a**) Dependencies between the segments have to be computed in the question analysis phase. **slow, low precision & recall**

Markov Logic Network

- **CASIA** (**he2014a**) Hard constraints like ILP, soft constraints flexibility **training needed low precision & recall**

Considering:

- Similarity of the phrase and the corresponding resource
- Popularity of a label for a resource
- Compatibility of the range and domain of a property with the arguments.

Xser (**xu2014a**) Solves ambiguity **fast, training needed**

Dissambiguation #9 - Summary

Which techniques to choose

ToDo

Query Construction #1 - Issues

Construct a **SPARQL** query that reflects user question and gets the answer.
Semantic Gap: Issues with how the information is encoded in the KB. One cannot deduce how the information is stored from the question.

"Which countries are in the European Union?"
Could be encoded as:
dbr:Greece dbp:member dbr:European_Union
dbr:France dbp:member dbr:European_Union
or as:
dbr:Greece dct:subj dbc:Member_states_of_the_European_Union
dbr:France dct:subj dbc:Member_states_of_the_European_Union
How to search correctly

2018-06-20

Approaches:

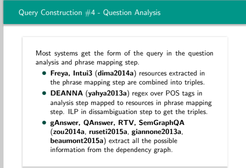
- Using templates
- Using information from the question analysis
- Using Semantic Parsers
- Using Machine Learning
- Using semantic information

Query Construction #3 - Templates

Templates with parts of the query to be filled, in general by triples.

- **QAKiS** (**cabrio2012a**) select queries with only one triple.
- **ISOFT** (**park2014a**) ASK over one triple, simple SELECT, COUNT and ORDER BY or FILTER.
- **PowerAqua** (**lopez2012a**) reduces the question to one or two triples (<= predicates).

Very restricted questions, language is too rich, disambiguity is key



- gAnswer: The graph takes the form of the final query, resources associated with nodes and edges are fetched from the KB and used in the query.
- QAnswer: Scan dependency tree to find subgraph tokens corresponding to resouces, many graphs, local dissambiguation to get the best ones. Top ranked graph chosen for query.
- RTV dependency graph -¿ ordered list of alternated properties and non properties. Resources searched and disambiguated with HMM.

Special mention to Xser (best results), next slide

Xser (**xu2014a**) 3 ML algorithms, two KB independant (on the question analysis phase), one KB dependant (on dissambiguation step)

- First algorithm: determines segments of the question corresponding to variables, properties, instances and classes.
- Second algorithm: find dependencies between phrases. (**Standford dependencies, PATTY**)
- Third algorithm: Dissambiguation with a **Structured Perceptron**

The problem with these methods is that they all **assume** that **is possible to deduce the structure of the SPARQL query from the structure of the question** without knowing how the knowledge is encoded in the KB.

2018-06-20

Core techniques of **QA Systems over KBs a Survey**

└─Query Construction

└─Query Construction #7 - Semantic Parsing

Compose a grammar and use it to extract structure from the query.

- **G**rammatical **F**ramework (**GFMed marginean2017a**)
- **F**eature-based **C**ontext **F**ree **G**rammar (**TR Discover**, **song2015a**)
- **C**ombinatorial **C**ategorial **G**rammar (**hakimov2015a**)
- **L**exical **T**ree **A**djoint **G**rammar (**TBSL unger2012a**, **BELA walter2012a**)

Query Construction #8 - Semantic Parsing

Question has to be well formulated. For each lexical item a corresponding semantic representation is needed. (ie married has to map with *dbo:spouse*). Learning corpus (**hakimov2015a**) or from POS tags (**unger2012a**). In general, **low recall**

Lexical item	Syntactic category	Semantic representation
Barack Obama	NP	$\texttt{dbr : Barack\_Obama}$
is	$(S\backslash NP)/(S\backslash NP)$	$\lambda f.\lambda x.f(x)$
married to	$(S\backslash NP)/NP$	$\lambda y.\lambda x.\texttt{dbo : spouse}(x, y)$
Michelle Obama	NP	$\texttt{dbr : Michelle\_Obama}$

Figure: Semantic parsed question *"Barack Obama is married to Michelle Obama"*

2018-06-20

Core techniques of **QA Systems over KBs a Survey**

└─Query Construction

└─Query Construction #9 - Machine Learning

CASIA (**he2014a**) (**low recall & precision**)

- Question Analysis step: extract features like position of a phrase and POS tags or the type of dependency in the dependency tree
- Phrase Mapping step: associate resources with phrase segments and extract more features
- Dissambiguation step: MLN with extracted features to find most probable relation between segments and most probable mapping. **retrained for each KB**

SINA (**shekarpour2015a**), **POMELO** (**hamon2014a**), **zhang2016a** do not rely on the syntactic features of the question, instead the whole process is done based on the KB, just with semantic information.

Advantages:

- **high recall, & precision**

Disadvantages:

- **computationally expensive**
- **does not respect user question syntax**. No difference between *"Who is the mother of Angela Merkel?"* and *"Angela Merkel is the mother of who?"*

Which techniques to choose?

2018-06-20

Core techniques of **QA Systems over KBs a Survey**

└─Conclusions

└─Conclusions

Thank you!

Questions?