Literature review
Natural Language Question Answering

Yassine Hamoudi

October 7, 2014

Introduction

Problematic

How answering natural language questions using existing structured databases ?

Introduction

Problematic

How answering natural language questions using existing structured databases ?

Objectives :

- question processing module : transform questions into normal form.
- databases processing module : find answers in databases.
- answer extraction module : return the exact answers, extracted after the previous step.

What we want to do :

- strong normalization of questions.
- searching answers in highly structured databases.
- full modular tool, to plug in easily as many databases as possible.

What we want to do :

- strong normalization of questions.
- searching answers in highly structured databases.
- full modular tool, to plug in easily as many databases as possible.

What we do not plan to do (?) :

- searching answers in not structured corpus of texts (newspapers, books...).
- trying to directly find sentences that best match with the question and probably contain the answer.

What we want to do :

- strong normalization of questions.
- searching answers in highly structured databases.
- full modular tool, to plug in easily as many databases as possible.

What we do not plan to do (?) :

- searching answers in not structured corpus of texts (newspapers, books...).
- trying to directly find sentences that best match with the question and probably contain the answer.

Warning

Most of the existing papers deal with the second kind of question answering. Their techniques cannot be directly applied to our subject.

Normal form representation

Most common representation : Subject Predicate Object (SPO)

Example

The turtle eats a salad.

$$\text{SPO} = (\text{turtle,eats,salad}) \text{ or } \text{eats}(\text{turtle,salad})$$

Normal form representation

Most common representation : Subject Predicate Object (SPO)

Example

The turtle eats a salad.

$$\text{SPO} = (\text{turtle,eats,salad}) \text{ or } \text{eats(turtle,salad)}$$

Expressing questions in first order logic :

- What is the birth date of the first president of the USA ?

  $\rightarrow \exists x \exists y$, be(x,first president of the USA) $\wedge$ wasBornIn(x,y)

Normal form representation

Most common representation : Subject Predicate Object (SPO)

Example

The turtle eats a salad.

$$\text{SPO} = (\text{turtle,eats,salad}) \text{ or } \text{eats}(\text{turtle,salad})$$

Expressing questions in first order logic :

- What is the birth date of the first president of the USA ?
  $\rightarrow \exists x \exists y$, be(x,first president of the USA) $\wedge$ wasBornIn(x,y)
- What is the capital of the southest African state ?
  $\rightarrow \exists x \exists y$, southestOf(x,Africa) $\wedge$ isCapitalOf(y,x)

Normal form representation

Most common representation : Subject Predicate Object (SPO)

Example

The turtle eats a salad.

SPO = (turtle,eats,salad) or eats(turtle,salad)

Expressing questions in first order logic :

- What is the birth date of the first president of the USA ?

  → $\exists x \exists y$, be(x,first president of the USA) $\wedge$ wasBornIn(x,y)
- What is the capital of the southest African state ?

  → $\exists x \exists y$, southestOf(x,Africa) $\wedge$ isCapitalOf(y,x)
- What is the name of the actress that played in Pocahontas and is married to a French violonist ?

  → $\exists x \exists y$, hasGender(x,woman) $\wedge$ playedIn(x,Pocahontas) $\wedge$ isMarriedTo(x,y) $\wedge$ hasNationality(y,French) $\wedge$ hasJob(y,violonist)

Finding the answer ⇔ finding a model in first order logic

- Each triplet conducts to quering a database :
  - → playedIn(x,Pocahontas) ↪ IMBd
  - → hasJob(y,violonist) ↪ MusicBrainz
  - → . . .
- Combining the answer to get the final result.
- More complex model : allowing universal quantification, negation...

RDF (Resource Description Framework)

- general framework for describing any Internet resource.
- a RDF document is a set of triplets (subject,predicate,object).
- `http://fr.wikipedia.org/wiki/Resource_Description_Framework`
- `http://www.w3.org/2001/sw/SW-FAQ#whrdf`

RDF (Resource Description Framework)

- general framework for describing any Internet resource.
- a RDF document is a set of triplets (subject,predicate,object).
- `http://fr.wikipedia.org/wiki/Resource_Description_Framework`
- `http://www.w3.org/2001/sw/SW-FAQ#whrdf`

SPARQL (SPARQL Protocol and RDF Query Language)

- an RDF query language.
- a W3C recommendation, fully standardized.
- can be used with a lot of knowledge bases.

Existing knowledge bases

- YAGO2 : more than 10 million entities and more than 120 million facts about these entities.
- DBpedia : 4.58 million entities, out of which 4.22 are classified in a consistent ontology.
- Freebase
- MusicBrainz
- **Wikidata**
- IMDb (Internet Movie Database)
- ...

→ most of them can be accessed via SPARQL queries (Wikidata ?).

→ more than 100 public SPARQL endpoints with dozens of billion of triples (`http://www.w3.org/wiki/SparqlEndpoints` for some examples).

→ more and more SPARQL endpoints in the future.

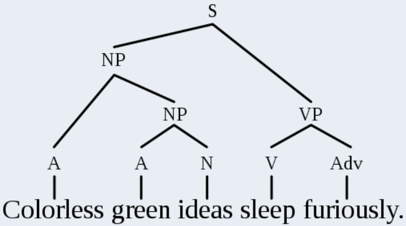
Changing our goals (?) :

- using SPARQL language (even if it is not the best tool to deal with wikidata ?).
- restricted modularity : only able to plug-in via SPARQL endpoint.
- designing a tool that deals with the wide range of SPARQL endpoints.

From syntax...

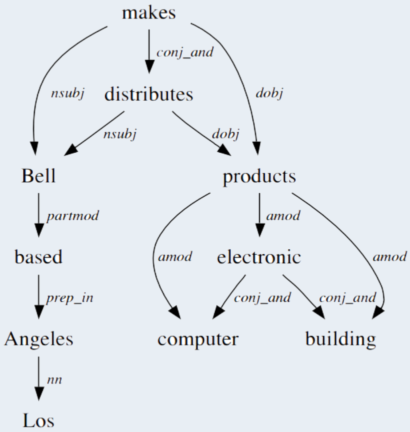
Parse structure tree (constituency relations)

Split the phrase according to its grammatical structure (noun phrase : NP, verb phrase : VP ...).



Dependency tree (dependency relations)

Reflect grammatical relationships between words in a sentence.



Bell, based in Los Angeles, makes and distributes electronic, computer and building products.

... to semantic

Parse structure tree

- not the best way to deal with semantic.
- an algorithm : `http://ailab.ijs.si/delia_rusu/Papers/is_2007.pdf`. Not very effective...

... to semantic

Parse structure tree

- not the best way to deal with semantic.
- an algorithm : http://ailab.ijs.si/delia_rusu/Papers/is_2007.pdf. Not very effective...

Dependency tree

- commonly used to perform triplet extraction.
- no good articles found on how to perform this.

... to semantic

Parse structure tree

- not the best way to deal with semantic.
- an algorithm : http://ailab.ijs.si/delia_rusu/Papers/is_2007.pdf. Not very effective...

Dependency tree

- commonly used to perform triplet extraction.
- no good articles found on how to perform this.

Other approachs :

- machine learning
- linear programming
- ...

→ usually a mix of heuristics (including parse structure/dependency tree)

Libraries

NLTK : http://www.nltk.org/

+ python
+ well documented, easy to use
- slow (according to many users)
- no statistical parser. Concretely : we cannot use it as is. Extra libraries :
  - http://stackoverflow.com/questions/6115677/english-grammar-for-parsing-in-nltk
  - http://stackoverflow.com/questions/14009330/how-to-use-malt-parser-in-python-nltk

Stanford Parser : http://nlp.stanford.edu/

- + well documented
- + faster than NLTK
- + frequently updated. A "state of the art" tool.
- + include a (the best ?) dependency parser : http://nlp.stanford.edu/software/dependencies_manual.pdf
- - java ?

Online demo :

- http://nlp.stanford.edu:8080/parser/index.jsp
- (coreNLP) : http://nlp.stanford.edu:8080/corenlp/process

Other tools : OpenNLP, Link Parser, Minipar, Berkeley Parser (online demo : http://tomato.banatao.berkeley.edu:8080/parser/parser.html)...

Treebanks

Text corpus with annotated syntactic (=structure) or semantic (=meaning) sentence structure.

Finding treebanks

- http://en.wikipedia.org/wiki/Treebank (existing tools)
- Question Treebank :
  http://www.computing.dcu.ie/~jjudge/qtreebank/ or http://nlp.stanford.edu/data/QuestionBank-Stanford.shtml

Semi-automatic / learning methods to build treebanks (?) :

- `http://www.hugo-zaragoza.net/academic/pdf/atserias_lrec10.pdf`
- `http://www.researchgate.net/publication/228739113_Semi-Automatic_Construction_of_a_Question_Treebank`

→ Mainly syntactic treebank (syntactic parse tree).
→ Some semantic treebanks (the most intereressant for machine learning ?).

Existing answering systems

Some tools :

- http://quepy.machinalis.com/
- https://www.youtube.com/watch?v=9v5nk1bzyD4
- http://www.ifi.uzh.ch/ddis/research/talking.html

Many other tools but source code not available.

Question Answering over Linked Data challenge :

→ http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/

→ 2013 winner : https://bitbucket.org/sebferre/squall2sparql (from Rennes)

Conclusion

- Lack of details about implementation in papers actually found.
- Most interesting papers (?) :
  - http://adapt.seiee.sjtu.edu.cn/~kangqi/qa.html : rewiew of 4 modern methods about question answering to databases.
  - http://people.mpi-inf.mpg.de/~myahya/papers/EMNLP2012_yahya.pdf
  - http://www.aifb.kit.edu/images/1/12/55540445.pdf
  - more on http://pad.aliens-lyon.fr/p/ppp-nlp
- Be aware of the difficulty of our task : very recent papers on question answering from knowledge bases claim no more than 30-50% of success.
- Relaxed problems :
  - interactions between the system and the user to find the answer.
  - restricted grammar for asking questions (not fully "natural question answering").

Keywords

question answering SPARQL RDF

natural language question answering

semantic parser subject verb object

predicate object subject

triple(t) extraction natural language RDF/SPARQL

natural language interfaces to databases

SVO (subject verb object)

translating questions into queries over knowledge base