Answering Live Questions from Heterogeneous Data Sources SMART in Live QA at TREC 2016

Edgard Marx and Sandro Coelho

SMART - AKSW, University of Leipzig, Germany
marx@informatik.uni-leipzig.de
http://smart.aksw.org

Abstract. A significant portion of information is today available in a digital format. However, users still face difficulties in accessing it. A big portion of the challenge consists in designing efficient approaches for reasoning over heterogeneous data sources. In this paper, we describe the participation of the Semantic Search and Question Answering group (SMART) in Live QA track at TREC 2016. SMART system answered live questions using information from Stackoverflow and DBpedia knowledge graph. SMART uses different approaches dubbed as `Cortex` for each different target data source and chose the answer based on the surface form's intersection with the given live question.

1 Introduction

The advances of technology in the information era have lead to the so-called Big Data. The Big Data can or not be structured and is continuously generated by us, humans, as well as from different type of smart devices. For instance, by today more than $10\,000$ *Resource Description Framework (RDF)*[1] datasets are public available.[2] However, for larger that the RDF data cloud seems to be, it still represents a small fraction of the data available on the Web. According to Web sites as WorldWebSize[3], there are more than 14 billions of Web pages on the Web. Although all this data is available, the biggest challenge, however, consists of helping users to access it. In this regard, Question Answering systems are being seen as one of the key technologies to overcome this obstacle.

In this work, we describe the participation of `SMART` system in Live QA track at TREC 2016. The Live QA track imposes an additional challenge, as the approaches used by the system must be either scalable and runtime efficient. This problem does not necessarily happen in other tracks as competitors can run their experiments in the background and later submit the results. This limitation didn't leave room to use more sophisticated techniques such as query expansion. Since the later can generate many answer candidates and thus, requires a higher computational complexity.

[1] `http://www.w3.org/RDF`
[2] `http://lodstats.aksw.org/`
[3] `http://www.worldwidewebsize.com/`

The remaining of this paper is organized as follows: Section 2 gives a summary of the system and its architecture. Section 3 describes the data sources and models used to retrieve the answer. Finally, Section 4 concludes giving an outlook to the future work.

2 System Overview

The SMART system is designed to perform reasoning over data from different sources. As data differs on its type (structured, unstructured, video and images), format (RDF, TEXT, MPEG, and JPEG) or information (Geographic, Life Science and Math), it does make sense to have different approaches to process it. To this extent, we implement what we call `Cortex`.

`Cortex` is an abstraction for data reasoner and it can use different approaches for question answering. For instance, in an RDF knowledge graph, the information is organized in facts and built upon the Resource Description Framework (RDF), but it can also be unstructured in other sources, e.g. `TEXT`. Furthermore, the information can have different characteristics. For example, the weather forecast needs to be height frequently updated in spite of other information such as a person's birth or death date. Moreover, the same information can be available in different sources and formats, following we enumerate some the data available (Figure 1):

- Knowledge Graphs: The information published in RDF format is increasing, but it is not the only format for publishing knowledge graphs. Moreover, knowledge graphs can include public, private as well as enterprise information. The challenge here consists in an efficient process the different information connected in the graph;
- Unstructured Data: Although knowledge graphs encompass a significant source of information, a big portion of the data is still in textual form. Be capable of processing this data can help users to access hidden contents;
- Logical Operations: There is a type of data that consist of math and logic operations, this data is processed in a different fashion than other kinds of information such as factual data;
- Temporal Data: A particular portion of the daily used data change very often—e.g. the weather forecast can change many times per day—and thus might be frequently updated. We differentiate this sort of data because it has a small life spanning time. Thus, they require handling in a different fashion.

3 Answering Live Questions

The SMART system is implemented using openQA framework [4] and Apache Lucene[4]. In this challenge, we implemented two `Cortexes`. One to process RDF Knowledge graphs and another to process unstructured data coming from Q&A forums. An RDF knowledge graph is defined in [3] as following:

[4] `lucene.apache.org`

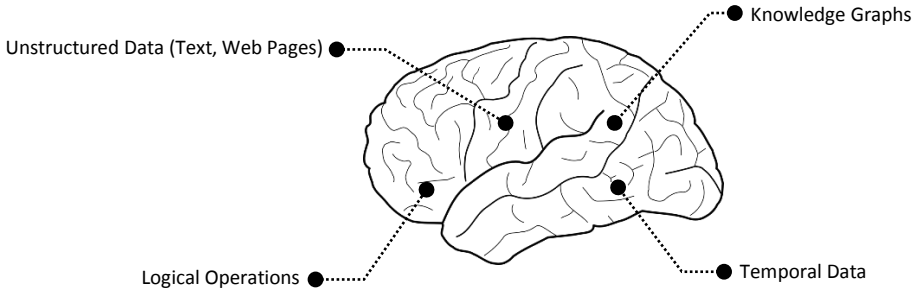

Fig. 1. Overview of the different `Cortexes` that can be used to process information from different data sourvces.

Definition 1 (RDF knowledge Graph, KG). *Formally, let K be a finite RDF knowledge graph (KG). K can be regarded as a set of triples $(s, p, o) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{P} \times (\mathcal{I} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{R} = \mathcal{I} \cup \mathcal{B}$ is the set of all RDF resources $r \in \mathcal{R}$ in the KG, $\mathcal{I}$ is the set of all IRIs, $\mathcal{B}$ is the set of all blank nodes, $\mathcal{B} \cap \mathcal{I} = \emptyset$. $\mathcal{P}$ is the set of all predicates, $\mathcal{P} \subseteq \mathcal{I}$. $\mathcal{L}$ is the set of all literals, $\mathcal{L} \subset \Sigma^*$ and $\mathcal{L} \cap \mathcal{I} = \emptyset$, where Σ is the unicode alphabet. $\mathcal{E}$ is the set of all entities, $\mathcal{E} = \mathcal{I} \cup \mathcal{B} \setminus \mathcal{P}$. An RDFTerm φ refers to any edge label $p \in P$ or vertex in the KG $\varphi \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$. A KG is modeled as a directed labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{D})$, where $\mathcal{V} = \mathcal{E} \cup \mathcal{L}$, $\mathcal{D} \subseteq \mathcal{E} \times (\mathcal{E} \cup \mathcal{L})$ and the labeling function[5] of the edges is a mapping $\lambda : \mathcal{D} \mapsto \mathcal{P}$. We disregard literal language tags and data types.*

To facilitate the information deployment and management, we made use of KBox [2]. In the following sections, we describe individually how each of the two `Cortexes` processes information and how the SMART system elect the most prominent hypothesis as possible answer.

3.1 Q&A Forums

In this challenge, the `Cortext` for Q&A forums used information extracted from Stackoverflow.[6] In Stackoverflow Q&A forum, each question represents a Web page. To enable the reasoning over the forum's content, we crawled the Stackoverflow Web site extracting the question and its corresponding height rated answer for each existing question page. After that, the extracted question/page was stored using Apache Lucene[7]. Each forum's question generated an entry containing three fields (the question, the answer, and its source). Only the question was indexed. Therefore, the entry could only be retrieved using question's words. That is, the best answer to a given live question is the one in which its corresponding Stackoverflow question achieves the highest Lucene `tf-idf` score. The Lucene `tdf-idf` function is formally defined as follows:

[5] Not to be confused with `rdfs:label`.
[6] `http://stackoverflow.com`
[7] `https://lucene.apache.org`

Definition 2 (Lucene tf-idf). *Given a query q and a document d, a score of a document is given by the function $score$ that receives the query and the document as parameter and computes the score of the document as follows:*[8]

$$\text{score}(q,d) = \text{coord}(q,d)\,\text{queryNorm}(q) \sum_{\forall t \in q} tf(t \in d) idf(t)^2 boost(t) norm(t,d)$$

The iner-functions of the above equation are defined as follows:

- $tf(t \in d)$ *correlates to the term's frequency, defined as the number of times term t appears in the currently scored document d;*
- $idf(t)$ *stands for Inverse Document Frequency;*
- $coord(q,d)$ *is a score factor based on how many of the query terms are found in the specified document;*
- $queryNorm(q)$ *is a normalizing factor used to make scores between queries comparable;*
- $boost(t)$ *is a search time boost of term t in the query q as specified in the query text, or as set by application, and;*
- $norm(t,d)$ *encapsulates a few (indexing time) boost and length factors.*

Notice that a document d in the Definition 2 corresponds to an indexed Stackoverflow's question.

3.2 RDF Knowledge Graphs

The `Cortex` operating into RDF knowledge graph was designed for answering fact-based questions. It processed information from DBpedia knowledge graph using the `⋆pah` approach [3]. The `⋆pah` approach works with a Semantic Weight Model (SWM) applied to a Term Network extracted from a structure called Semantic Connected Component (SCC). The Term Network, SCC and SWM are formally defined in [3] as follows:

Definition 3 (Term Network). *A Term Network is a graph whose vertices are labeled with terms.*

Definition 4 (Semantic Connected Component). *The Semantic Connected Component (SCC) of an entity e in an RDF graph G under a consequence relation $\models$ is defined as $SCC_{G,\models}(e) \coloneqq \{(e,p,o) \mid G \models \{(e,p,o)\}\} \cup \{(p,$ rdfs:label$,l) \in G\} \cup \{(o,$ rdfs:label$,l) \in G\}\}$. If the graph and consequence relation are clear from the context, we use the shorter notation $SCC(e)$.*

Definition 5 (Semantic Weight Model (SWM)). *Each* token t *in $T(q)$ is first mapped to the* paths *of the* SCC S*. The set of matched tokens from a path γ is returned by the*

[8] The given information bellow is an excerpt of the full text found in TDFIDFSimilarty page at `https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`.

function $TP(\gamma, q)$. *A* path match *of an* SCC S *is evaluated by the function* $\mathrm{MTP}(\gamma, q, S)$ *using a path weighting function* $\mathrm{w} : D^+ \rightarrow R$.

$$TP(\gamma, q) \coloneqq \{t \in T(LP(\gamma)) \mid \exists t' \in T(q) : \delta(t, t') < \theta\}$$
$$\mathrm{MTP}(\gamma, q, S) \coloneqq \{t \in TP(\gamma, q) \mid \forall \gamma' \in D(S)^+ : \mathrm{w}(\gamma)|TP(\gamma, q)| \geq \mathrm{w}(\gamma')|TP(\gamma', q)|\}$$

The final score of an SCC S *is a sum of its* n path-scores *and is measured by the function* $\mathrm{score}(S)$*, as follows:*

$$\mathrm{score}(S) = \sum_{\gamma \in D(S)^+} \begin{cases} \mathrm{w}(\gamma)|TP(\gamma, q)| & \textit{if } \mathrm{MTP}(\gamma, q, S) \neq \emptyset, \\ 0 & \textit{otherwise.} \end{cases}$$

In case there are terms matching multiple paths and the paths have an equal number of matched terms and equal score, only one of the path scores is added to the SCC *score.*

3.3 Answering

As `Cortexes` can diverge in content and approaches, they can generate different answers for a given question. In this work, we choose to select the approach that covers the biggest number of words in the question's surface form.

In the Q&A forum's `Cortex`, we check the surface form of the answer's question whereas in `Cortex` for the knowledge base, the surface form of the SCC graph. Answers coming from knowledge bases or Q&A forums can contain quality problems. In case there was a tie between the surface forms of the data processing from different `Cortexes`, the answer from Q&A forum was used. This approach was performed due to our intuition that Q&A forum contains a more precise answer than knowledge bases. The reason is that our `Cortex` operating on RDF knowledge base is not designed to give full answers to questions. Furthermore, our hypothesis is that answers from Q&A forums have better quality since our system uses the forum voting system to filter inaccurate answers.

4 Conclusion, Limitations & Future Works

In this work, we presented the approaches used by the SMART system on Live QA track of TREC 2016. There are a few remaining challenges that we plan to address in future implementations such as (i) the treatment of complex queries [1] as well as (ii) terms with different forms (query expansion), (iii) the addition of other `Cortexes` to deal with other sources of information, and (iv) investigate other methods for the election of the most prominent answer. In future work, we plan to address the mentioned challenges.

Acknowledgements This work was supported by a grant from the EU H2020 Framework Programme provided for the projects Big Data Europe (GA no. 644564), HOBBIT (GA no. 688227), and CNPq under the program Ciências Sem Fronteiras.

References

1. Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., Ngonga Ngomo, A.C.: Survey on challenges of Question Answering in the Semantic Web. Submitted to the Semantic Web Journal (2016)
2. Marx, E., Baron, C., Soru, T., Auer, S., Ngomo Ngonga, A.C.: KBox – Transparently Shifting Query Execution on Knowledge Graphs to the Edge. In: submited to ICSC (2016)
3. Marx, E., Höffner, K., Shekarpour, S., Ngomo Ngonga, A.C., Lehmann, J., Auer, S.: Exploring Term Networks for Semantic Search over RDF Knowledge Graphs. In: 10th International Conference on Metadata and Semantics Research. MTSR (2016)
4. Marx, E., Usbeck, R., Ngomo Ngonga, A.C., Höffner, K., Lehmann, J., Auer, S.: Towards an Open Question Answering architecture. In: SEMANTiCS (2014)