Revealing the "important" properties of Entities

Ghislain A. Atemezing[1], Ahmad Assaf[1], Raphaël Troncy[1] and Elena Cabrio[12]

[1] EURECOM, Sophia Antipolis, France,
<atemezin@eurecom.fr>
[2] INRIA, France, <elena.cabrio@inria.fr>

Abstract. In knowledge bases and more precisely in the Web of Data, entities have a lot of properties. A quick view to the different versions of DBpedia can give an idea of the phenomenon. However, it is still difficult to decide which ones are important than others depending on how we want to use these entities, such as for a visualization of some basic facts about the given entity. In this paper, we perform reverse engineering on the Google Knowledge graph panel to find out what are the properties shown according to the type of the entity. We compare the results obtained with users surveyed on some Entities. The preliminary results are promising as they shape the path towards a recommendation tool for detecting the core properties important to Entities.

Keywords: Crowdsourcing, Google Knowledge panel, visualization, scrapping, knowledge elicitation, intrinsic properties

1 Introduction

{TODO: rewrite this**}** - 1) Motivation: in knowledge bases, entities have a lot of properties. Deciding which ones are more important than others depending on how we want to use these entities. Two use cases: . a) visualization of some basic facts about entities, for a multimedia QA system (QakisMedia) or for a second screen application (LinkedTV) . b) data integration (ontology matching), those properties having a bigger weights when computing alignments - 2) Approach 1: Google knowledge graph panel reverse engineering ... algorithms + first results - 3) Approach 2: User survey ... setup + results analysis - 4) Vocab for representing those "important" properties and dataset publication

2 Related Work

{TODO: @summarize the work of Thomas, Michiel Hildebrand, etc**}**

3 Google Panel Reverse Engineering

Algorithm 1 Google knowledge graph panel reverse engineering Algorithm

```
INITIALIZE $equivalentClasses(DBpedia, Freebase)$ AS $vectorClasses$
Upload $vectorClasses$ for querying processing
Set $n$ AS number-of-instances-to-query
for each $conceptType \in vectorClasses$ do
    SELECT $n$ instances
    $listInstances \leftarrow$ SELECT-SPARQL($conceptType$, $n$)
    for each $instance \in listInstances$ do
        CALL http://www.google.com/search?q=$instance$
        SCRAP GOOGLE KNOWLEDGE PANEL
        $gkpProperties \leftarrow$ GetData(DOM, EXIST(GKP))
    end for
    COMPUTE ocurrences for each $prop \in gkpProperties$
end for
return $gkpProperties$
```

4 User Survey Settings

5 Preliminary Results

6 A Vocabulary for representing important properties

7 Conclusion and Future Work

Acknowledgments

This work has been partially supported by the French National Research Agency (ANR) within the Datalift Project, under grant number ANR-10-CORD-009.