RDF Knowledge Graph Visualization From a Knowledge Extraction System

Fadhela Kerdjoudj[1,2] and Olivier Curé[1]

[1] Université Paris-Est Marne-la-vallée, LIGM, CNRS UMR 8049, France.,
[2] GEOLSemantics, 12 rue Raspail 94250, Gentilly, France.
fadhela.kerdjoudj,ocure@u-pem.fr,

Abstract. In this work, we present a system to visualize RDF knowledge graphs. These graphs are obtained from a knowledge extraction system designed by GEOLSemantics. This extraction is performed using natural language processing and Trigger Detection. The user can visualize subgraphs by selecting some ontology features like concepts or individuals. The system is also multilingual, with the use of the annotated ontology in Arabic, Chinese, English and French.

1 Introduction

During the last decades, many knowledge extraction systems have emerged aiming at automating textual document processing. The importance of these systems is highlighted by the proliferation of textual publications on the web,*e.g.* social media, blogs or journals. In order to extract as much relevant information as possible, it is necessary to exploit the power offered by the semantic web and its related technologies. These technologies, namely, vocabularies (RDF, OWL, SKOS...), query language (SPARQL), inference services, Linked Open Data (LOD), allow to represent, access, reason over and interconnect extracted data. To obtain these data, we use a knowledge extraction system based on Natural Language Processing (NLP) to populate a Knowledge Base.

In this article we present a component of this system which deals with RDF knowledge graph visualization. It allows to build subgraphs by selecting either ontology concepts or individuals. Indeed, the size of the knowledge graph extracted is proportional to the length of the text. Therefore, the graph could become fairly large and dense. To deal with this issue, we propose an approach that helps visualizing and summarizing the extracted knowledge.

2 Knowledge extraction system

The Web contains a huge number of documents from heterogeneous sources like forums, blogs, tweets, newspaper or Wikipedia articles. However, these documents cannot be used directly by programs because they are mainly intended for humans. Before the emergence of the Semantic Web, only human beings could access the necessary background knowledge to interpret these documents.

The extraction and representation framework developed at GEOLSemantics, a french NLP start-up, uses some technologies of the Semantic Web and organizes its extraction process in the following three steps:

2.1 Deep Morphosyntactic Analysis

The deep morphosyntactic analysis consists of the following steps.

- Text Indentation: the text is splitted into tokens using regular expressions which allows to identify capital letters, numbers, dates, *etc.*
- Morphological Processing : the aims is to recover linguistic token information, it allows to correct some writing errors such as dashes, to identify idiomatic expressions such as: *lose yourself, swing into action, cut the cackle.*
- Named entity recognition : the named entities, namely: Person, Organization, Location are identified using two methods: *(i)* Thesaurus consultation, based on LOD such as DBpedia and Geonames. *(ii)* Declarative rules based on announcers, such as President, Mister, city, airport, *etc.*
- Syntactic Analysis : allows to represent the syntactic structure of a text. It indicates how the grammatical categories are arranged, *e.g.*; noun-verb-adjective. For instance, in sentence matching a Subject Verb Object structure, the verb (in active form) indicates the action, the subject allows to identify the agent of the action performed on the object.
  Indeed, some other processes are performed like: transform passive forms to active, resolve anaphora, detect negation and verb tense which gives information about the modalities of the action.

2.2 Knowledge Extraction

The knowledge extraction allows to identify the named entities and the relations between them. This is performed using an ontology-based approach which defines the different concepts needed for annotating entries of the original text. Each extracted concept is associated to a list of rule patterns which help in structuring the knowledge base to be extracted. All these processes are organized as follows:

- Probable Concept Selection: It consists in spotting the trigger. It can be a word, an expression or a relation. Each trigger is associated to an ontology class and a number of rules.
- Rule Selection: Each trigger is associated to a list of rule patterns. From the different relations identified in the syntactic analysis a matching approach enables to select the most relevant pattern.
- Triple Creation: the rule selected helps to create triples using the given rule pattern. Then the result is structured as RDF triples where the concept indicates the subject, the relation is the predicate and the concept related to or the attribute indicates the object of the triple.

2.3 Integration

In this step, our aim is to bring more consistency to the extracted knowledge by performing the following processes: *(i)* Coreference resolution: group all the instances of each entity. *(ii)* Relative dates resolution: transform all relative dates like *today, last week* to absolute dates. *(iii)* Complete the extraction with implicit information which can be inferred when reading a text such as the date or the place. *(iv)* Label creation: following the token positions indicated by the morphosyntactic processing, the labels are retrieved from the original text.

3 Visualization Features

3.1 Ontology description

Using an RDF triple based representation and ontology description, the text can be represented as a knowledge graph which contains all needed information.
Currently our ontology contains a fex hundred of classes and properties. It is regularly enriched to support more concepts and domains. The classes considered are mainly :

- Named entities namely person, organization, location, measure, date.
- Facts such as professional experience, studies, family relation, personal relation, event relation, organization relation.
- Events like meeting, movement, violent act, conviction, appointment, arrest.

The object properties describe the relation between entities such as the address of a person or an organization, the date and the place where an event takes place. Finally the datatype properties are literals which describe mainly the named entities such as names, types, values.
We would like to stress that a great importance has been given to the ontology design. All the classes and properties have to be labeled in different languages. At this point, our ontology contains Arabic, French, English and Chinese. Also, all the properties must be related to their respective domain and range.

3.2 Graph features

The knowledge extracted comply with the ontology description, the triples can then be related to each other and the graph can be constructed. In our graph representation, instead of using URIs to denote nodes, we use icons and labels. This allows the user to spot the requested information in easier way than reading all the URIs which are usually less illustrative. The edges are also denoted with labels as they were indicated in the ontology.

Multilingual aspect: Defining the labels in the ontology in different languages allows then to visualize the graph in a multilingual form as shown in Figure 1 which describes the knowledge extracted from Example 1.

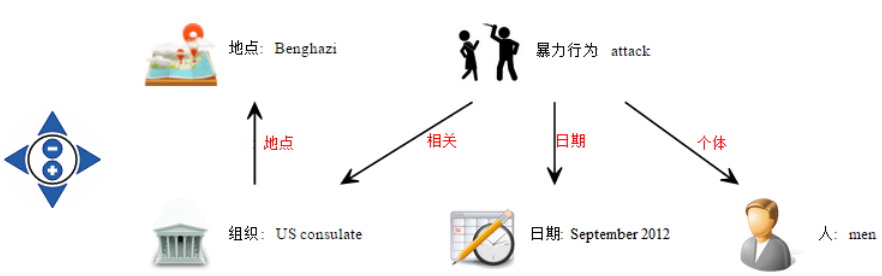

Fig. 1. RDF graph of the example 1

Example 1. In September 2012, the US consulate in Benghazi was attacked by armed men.

Literals could also be translated into the selected language. For the sake of clarity, we keep them in their original form (*i.e.* like they were cited in the text).

Faceted search: As we have already stated, the graph can be dense and hardly understandable, here we propose a selection of subgraphs which can help the user to directly visualize the information need. Two selections are proposed :

1. **Concept Selection:** The RDF is parsed in order to retrieve all the classes instantiated in the viewed text. Hence, we select all the rdf:type that the RDF contains. For instance, Person, Location, Organization, ViolentAct and Date in Example 1.
2. **Instance Selection:** Here we propose all the nodes extracted from the text. Labels are used to help the user to select the one that he wants. In Example 1, the instances are: Benghazi, attack, man, September 2012, US consulate.

In the latter, the user can select the graph degree depth. It indicates how deep the subgraph must be, *i.e.* if adjacent nodes need to be developed.
To avoid cluttering, we only display relations which denote object properties. Datatype properties are viewed when hovering a node as tooltips as shown in Figure 2.

Table View: We also propose a table view of the triples extracted. The first column contains the Subject, the second, the predicate and the third one, the Object.

4 Implementation

The visualization module of our system is a web interface developed in Java. The graph is built using GraphViz[3]. Graphviz [2] constructs a graph from an

[3] http://www.graphviz.org/Documentation.php

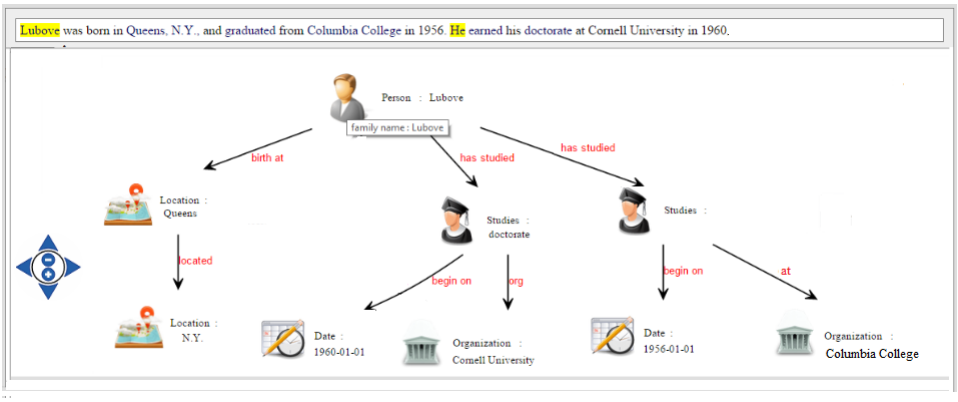

Fig. 2. Graph with tooltip and link with text.

entry in DOT [3] language. The diagrams are rendered in different formats: PNG, PDF, SVG, *etc.* The text (labels) can be handled via useful features such as: font, color, size, hyperlinks, custom shapes. In addition, the layout can be hierarchical, radial or circular. Moreover, we used some javascript code to link the graph to the text by highlighting the trigger when hoovering the corresponding node. Finally, the ontology is parsed with the Jena API [1], to retrieve classes and properties, hierarchies and annotations.

5 Conclusion and future work

In this work, we present a system to visualize an RDF knowledge graph. It allows to select subgraph, which is especially useful in the case of big graphs obtained from long text processing. we also explain the role played by the ontology in the visualization, it helps to provide more clarity and provide a multilingual interpretation of the text. As future work, we prospect to handle more RDF extractions such as Yago and DBpedia, and make the graph more interactive by allowing to move the nodes for instance.

References

1. Jeremy J Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83. ACM, 2004.
2. John Ellson, Emden R Gansner, Eleftherios Koutsofios, Stephen C North, and Gordon Woodhull. Graphviz and dynagraphstatic and dynamic graph drawing tools. In *Graph drawing software*, pages 127–148. Springer, 2004.
3. Eleftherios Koutsofios, Stephen North, et al. Drawing graphs with dot. Technical report, Technical Report 910904-59113-08TM, AT&T Bell Laboratories, Murray Hill, NJ, 1991.