Extracting Data from Text and Querying it via KGQA and BESt Queries

Maurizio Atzori

[1]*Department of Mathematics and Computer Science, University of Cagliari, Italy*
[2]*Big Data Lab, National Interuniversity Consortium for Informatics (CINI), Rome, Italy*

Abstract

This paper will review the progresses done at the University of Cagliari on the exploitation of free text corpora in order to extract structured information that can be then queried using both standard and advanced querying techniques. Unsupervised techniques to induce knowledge graphs (entity, relations, ontological hierarchies) from untagged text, including ad-hoc tasks (such as set expansion, relation extraction, etc.) available in our python library *OKgraph* will be discussed. We will also discuss advanced techniques that have been developed to query such structured data, including the use of natural language via Knowledge Graph Question Answering (KGQA) and the so-called By-Example Structured (BESt) Queries, developed in collaboration with UCLA.

Keywords

Information and Data Extraction, Entity Linking, Natural Language Processing, Question Answering, Knowledge Graphs

1. Introduction

Data – meaning structured information that can be used in formal queries – usually associated with metadata to interpret values, is fundamental for most information processes to tackle user information needs. As a simple but popular example, we all realized that keyword-based search engines in the last years evolved, often providing direct answers in form of structured data taken from their knowledge bases. These source of structured information, that are internally stored and represented in form of graphs – and therefore called knowledge graphs (KG) – are very valuable but unfortunately also difficult or expensive to create from scratch. In fact, although in some domains a number of public KG can be found, in many other applications (e.g., those about internal company knowledge such as products, documentation, etc.) they may not be available, and have to be curated by a usually expensive human-driven process.

At University of Cagliari we approached the problem by developing a set of tools that can help users to extract structured information taking advantage from existing texts that may be already available. These NLP tools, contained in the OKgraph library, allow users to extract

ITADATA2022: The 1st Italian Conference on Big Data and Data Science, September 20–21, 2022, Milan, Italy
*This work is partially funded by projects PRIN 2017 (2019-2022) *HOPE* and Fondazione di Sardegna *ASTRID* (CUP F75F21001220007).
atzori@unica.it (M. Atzori)
0000-0001-6112-7310 (M. Atzori)

many structured information that can be then used to form a KG from any running text (such as documents or other texts). This line of research is described in Section 2.

On the other hand, once users have access to a Knowledge Graph, either automatically extracted with tools like OKgraph or downloaded from an available source, their information needs are not necessarily satisfied yet. In fact, answers may be difficult to find in KG, requiring, e.g., data joins, filtering, and semantic analysis over often thousands of potential properties and sometimes millions of concepts appearing in these KG. Therefore, another line of research that we are following is focused on how to simplify access to KG via user-friendly query methods. In Section 3 we describe two different methods that can help casual users to pose formal queries against knowledge graphs.

The former can use the system to release semantically-annotated and high-quality open data, while the latter can access such data in a user-friendly fashion.

The work in this paper forms part of a wider PRIN research project called *High-quality Open data Publishing and Enrichment (HOPE)*[1], whose main goal is the development of a web-based open data management system addressed to public and private organizations [1].

2. Extracting Structured Information via the Open Knowledge Graph Library (OKgraph)

OKgraph is a python3 library developed at University of Cagliari in order to extract structured (ontological) data out of free text [2]. It has been designed with a few desiderata in mind:

- language independent (any space-separated natural language as input)[2]
- fully unsupervised (only unlabeled text as input), that is without exploiting supervised methods or models created using supervised methods
- only free (running) text as input (not semistructured such as html/xml or other structures)

The above requirements are quite challenging, but necessary to address the research question behind OKgraph: *how much structured information and data can be extracted from running text without supervision?*

The high-level architecture of the library is shown in Fig. 1. It expects a large corpus as input from which unsupervised models are computed, in particular word embeddings. Optionally, pre-computed models can be provided as input if available. Word embeddings are used in order to exploit their geometrical properties (see Fig. 2), embedding semantics that can be represented in the form of a Knowledge graph. This is the main approach that OKgraph follows in order to address the NLP tasks associated to unsupervised Knowledge Graph extraction.

Once embeddings are available, OKgraph exploits them to address different Natural Language Processing tasks that are useful to represent an ontology/knowledge graph of concepts.

In the following we details the main tasks that have been addressed.

[1] http://hope-prin.org

[2] Scriptio-continua corpora and languages needs third-party tokenization techniques, known as word segmenters (see e.g. https://github.com/tkng/micter for an implementation based on Support Vector Machines)

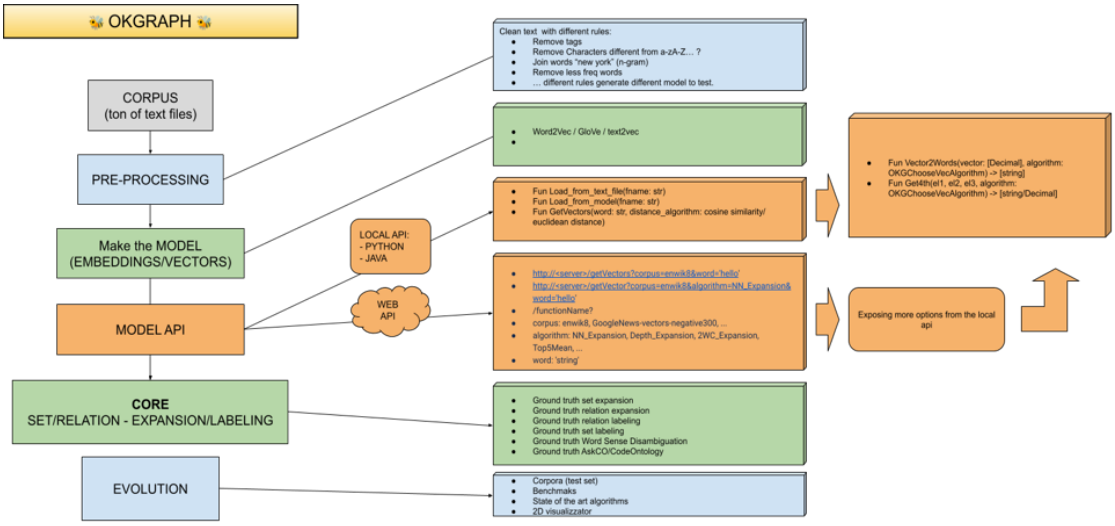

Figure 1: Architecture of the emphOKgraph python library for information extraction. (https://github.com/atzori/okgraph).

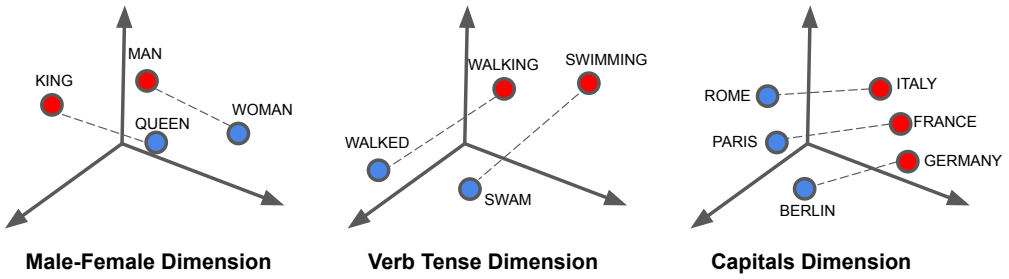

Figure 2: An example of how word embeddings and their distribution in space. Semantic relations can be extracted by exploiting the vector space in an unsupervised way.

2.1. Set Expansion

Set expansion is an NLP task that given one or a short set of words, the algorithm continues this set with a list of other "same-type" words (also known as *co-hyponyms*). For instance, given a set of 3 Italian cities such as `Milan`, `Rome`, `Bari`, OKgraph is able to provide a list of other Italian cities, such as `Turin`, `Palermo`, `Venice`, etc.

Solving this task is very useful to populate a knowledge graph with "sibling" instances, all of them belonging to the same semantic class.

Okgraph performs set expansion by means of word embedding semantic similarities and analysing geometrical directions of word vectors [2, 3]. Since sibling relation is transitive, so is

expected in the semantic similarities of good candidates of set expansion. Candidate that are outliers to previously-selected output (i.e., seeds) are therefore discarded.

2.2. Set Labeling

With Set Labeling, we mean the following problem: given one or a short set of words, returns a list of short strings (labels) describing the given set (its type or *hyperonym*).

OKgraph provides some heuristics to extract these labels, that can be used in the context of an Ontological Graph as the parent of same-type instances. For instance given a set of countries such as `Italy`, `France`, `Germany`, OKgraph can provide some candidate labels such as "country", "nation", (etc.) in a fully unsupervised way [4]. The idea is that some words appear more often in contexts where seed words appear, e.g. "country" when also "Italy" or "France" or "Germany" are present. Hearst or other similar *is-a* patterns can also be used if available.

2.3. Relation Expansion

Structured (ontological) information is usually represented by means of graphs. These are formed by nodes, for which the previous tasks provides useful insights, and edges, usually represented using pair of nodes.

Relation expansion try to address the problem in which, in the context of a Knowledge graph curation, we want to expand it finding new edges out of a running text. Basically, given one or a short set of word pairs, in this task OKgraph continues this set with a list of tuples having the same implicit relation of the given tuples.

For instance by providing pairs of nations and corresponding capitals, such as `Italy&Rome`, `France&Paris`, the library may find solutions such as `Germany&Berlin`, `Spain&Madrid`.

This is obtained by exploiting set expansion over the first element of pairs and the second element of pairs, and then finding the best matches between results to form other pairs.

2.4. Relation Labeling

In Relation Labeling, given one or a short set of word tuples, we expect a list of short strings (labels) describing the implicit relation of the tuples in the given set. For instance, given `Italy&Rome`, `France&Paris`, `Germany&Berlin`, we may expect a description such as the string "capital".

This is therefore very related to Set Labeling but applied to pairs of concepts/words. It can be solved in the same way as Set Labeling, but limiting search space on contexts where both words in the pair are present.

In the context of Knowledge Graph Extraction, this NLP task would help to assign labels to edges extracted, e.g., via relation extraction.

3. Querying Structured Information

Whenever we have access to a structured information such as a Knowledge Graph, either because extracted (e.g., via NLP methods such those provided by OKgraph), or because available in the first place, the data is really useful only if it can be exploited and queried in a easy, user-friendly way.

At the University of Cagliari, in collaboration with the research group of Prof. Carlo Zaniolo at University of California, Los Angeles, we followed two distinct approaches:

1. Knowledge Graph Question Answering (KGQA), that is formulating the information need using natural language and then translating it automatically to a structured query language (such as SPARQL)
2. By-Example Structured Queries (BEStQ), a novel method to easily query Knowledge graphs, such as DBpedia, using simple editable "Infoboxes" (synoptic tables)

3.1. Question Answering over Knowledge Graphs (KGQA)

Question Answering over Knowledge Graph is the problem of translating a user question posed in natural language into a formal query language, typically SPARQL. A popular event in the field is the QALD (Question Answering over Linked Data) Challenge, that allows researchers to compare and discuss their results on this task using common benchmarks. In the context of a QALD challenge event, we developed QA^3 (read as "Q-A-cube") [5], a question answering system that answers statistical questions by generating the corresponding SPARQL query that can be run over a LinkedSpending endpoint to obtain the correct results.

The system showed good performance over the QALD benchmark for LinkedSpending data [6], made possible thanks to its 3-step process described in Fig. 3.

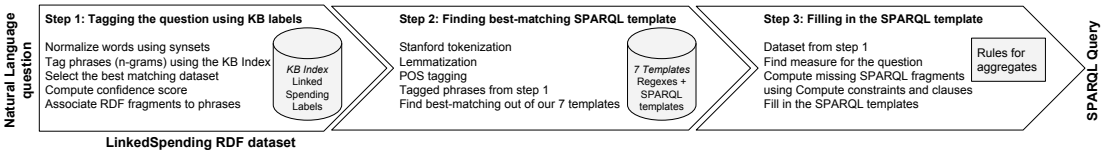

Figure 3: The 3-step process that QA^3 follows in order to generate the SPARQL query over LinkedSpending.

3.2. By-Example Structured Queries (BEStQ)

The user-friendly By-Example Structured (BESt) Query interface has been developed in the context of SWiPE [7, 8], whereby simple conditions entered in the property fields of the InfoBoxes, are turned into a SPARQL query executable on DBpedia. SWiPE is the integration of the BEStQ approach [9] to Wikipedia Infoboxes in order to query the DBpedia Knowledge Graph.

In general, the BEStQ approach extends the Query-by-Example approach from relational databases to let users enter constraints in the property fields of the InfoBoxes, which are now turned into active forms accepting query conditions that can also be complemented with the

Figure 4: SWiPE shows the By-Example Structured Queries applied to a Wikipedia Infobox: fields Region and mayor have been overwritten with some user constraints, composing a query.

keyword-based search conditions [10]. Thus in SWiPE the question of finding people named massimiliano which are also mayors of tuscany cities can be answered easily (see Fig. 4 and Fig. 5) and so are powerful queries involving conditions, joins and even aggregates, that can be entered in this way and combined with the traditional keyword-based searches on Wikipedia.

Indeed, SWiPE provides a unified user-friendly system to answer the simple requests that are typical of Question Answering along with the more complex ones that require the power of SPARQL or other structured query languages.

4. Conclusions

In this paper we have reviewed two research lines followed at University of Cagliari that helps user gaining more value from untagged textual data (in order to extract structured data) and then querying them in a user-friendly way. Our future work is focused on improving the extraction phase by integrating the KG generation inside the OKgraph library and creating an hybrid approch for querying structured data, based on BEStQ with editable field that accepts natural language constraints as input, as it happens in question answering but in the context of the field at hand.

Acknowledgements

This work is partially supported by PRIN 2017 (2019-2022) project *HOPE - High quality Open data Publishing and Enrichment* (http://hope-prin.org/) and Fondazione di Sardegna project *ASTRID* -

Figure 5: The results of the structured query formulated in Fig. 4.

Advanced learning STRategies for high-dimensional and Imbalanced Data (CUP F75F21001220007). The author wishes to acknowledge the precious work of Prof. Carlo Zaniolo (UCLA) described in Section 3.

References

[1] D. Marcia, M. Sanguinetti, M. Atzori, User-friendly query interfaces for the HOPE project, in: V. W. Anelli, T. D. Noia, N. Ferro, F. Narducci (Eds.), Proceedings of the 11th Italian Information Retrieval Workshop 2021, Bari, Italy, September 13-15, 2021, volume 2947 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2947/paper26.pdf.

[2] M. Atzori, S. Balloccu, A. Bellanti, E. Mameli, S. R. Usai, Okgraph: Unsupervised structured data extraction from plain text, in: M. Agosti, E. D. Buccio, M. Melucci, S. Mizzaro, G. Pasi, F. Silvestri (Eds.), Proceedings of the 10th Italian Information Retrieval Workshop, Padova, Italy, September 16-18, 2019, volume 2441 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 30–31. URL: http://ceur-ws.org/Vol-2441/paper19.pdf.

[3] M. Atzori, S. Balloccu, A. Bellanti, Unsupervised singleton expansion from free text, in: ICSC 2018, IEEE Computer Society, 2018, pp. 180–185. doi:10.1109/ICSC.2018.00033.

[4] M. Atzori, S. Balloccu, Fully-unsupervised embeddings-based hypernym discovery, Inf. 11 (2020) 268. doi:10.3390/info11050268.

[5] M. Atzori, G. M. Mazzeo, C. Zaniolo, QA 3 : A natural language approach to question answering over RDF data cubes, Semantic Web 10 (2019) 587–604. URL: https://doi.org/10.3233/SW-180328. doi:10.3233/SW-180328.

[6] K. Höffner, M. Martin, J. Lehmann, LinkedSpending: OpenSpending becomes Linked Open Data, Semantic Web Journal (2015). URL: http://www.semantic-web-journal.net/system/files/swj923.pdf. doi:10.3233/SW-150172.

[7] M. Atzori, C. Zaniolo, Swipe: searching wikipedia by example, in: A. Mille, F. Gandon, J. Misselis, M. Rabinovich, S. Staab (Eds.), Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume), ACM, 2012, pp. 309–312. URL: https://doi.org/10.1145/2187980.2188036. doi:10.1145/2187980.2188036.

[8] A. Dessi, A. Maxia, M. Atzori, C. Zaniolo, Supporting semantic web search and structured queries on mobile devices, in: R. D. Virgilio, J. Geller, P. Cappellari, M. Roantree (Eds.), 3RD International Workshop on Semantic Search over the Web, SSW '13, Riva del Garda, Italy, August 30, 2013, ACM, 2013, pp. 5:1–5:4. URL: https://doi.org/10.1145/2509908.2509910. doi:10.1145/2509908.2509910.

[9] H. Mousavi, M. Atzori, S. Gao, C. Zaniolo, Text-mining, structured queries, and knowledge management on web document corpora, SIGMOD Rec. 43 (2014) 48–54. doi:10.1145/2694428.2694437.

[10] H. Mousavi, M. Atzori, S. Gao, C. Zaniolo, Text-mining, structured queries, and knowledge management on web document corpora, SIGMOD Rec. 43 (2014) 48–54. URL: https://doi.org/10.1145/2694428.2694437. doi:10.1145/2694428.2694437.