Knowledge-Based Systems 108 (2016) 33–41

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Identifying motifs for evaluating open knowledge extraction on the Web

Aldo Gangemi[a,b,*], Diego Reforgiato Recupero[b,c], Misael Mongiovì[b], Andrea Giovanni Nuzzolese[b], Valentina Presutti[b]

[a] Université Paris 13, Sorbonne Cité, CNRS, Paris, France
[b] ISTC-CNR, via Gaifami, 18, 95126 Catania, Italy
[c] Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy

ARTICLE INFO

Article history:
Received 15 November 2015
Revised 10 May 2016
Accepted 11 May 2016
Available online 12 May 2016

Keywords:
Machine reading
Knowledge extraction
RDF
Semantic web
Linked open data

ABSTRACT

Open Knowledge Extraction (OKE) is the process of extracting knowledge from text and representing it in formalized machine readable format, by means of unsupervised, open-domain and abstractive techniques. Despite the growing presence of tools for reusing NLP results as linked data (LD), there is still lack of established practices and benchmarks for the evaluation of OKE results tailored to LD. In this paper, we propose to address this issue by constructing RDF graph banks, based on the definition of logical patterns called *OKE Motifs*. We demonstrate the usage and extraction techniques of motifs using a broad-coverage OKE tool for the Semantic Web called FRED. Finally, we use identified motifs as empirical data for assessing the quality of OKE results, and show how they can be extended trough a use case represented by an application within the Semantic Sentiment Analysis domain.

1. Introduction

Translating natural language text to formal data that can be used or integrated into knowledge bases is an important research task due to its applications in intelligent systems and data science, and therefore it is central to the Semantic Web (SW) community. One of the main open challenges is to establish shared practices and benchmarks for evaluating its results.

In recent years, the production of structured data from text has become scalable. Machine reading is a good example. In [1], the machine reading paradigm is defined as a procedure for extracting knowledge from text by relying on bootstrapped, self-supervised Natural Language Processing (NLP) performed on basic tasks. Machine reading can process massive amounts of text in reasonable time, can detect regularities hardly noticeable by humans, and its results can be reused by machines for applied tasks. The same techniques can be combined with logic-oriented approaches in order to produce formal knowledge from text, i.e., to perform OKE, which can be defined as *the extraction of knowledge from text, and its representation in formalized machine readable form* (see [2] for a survey on web data extraction and [3] for the work that introduced OKE). OKE is unsupervised, open-domain, and abstractive.[1] A key problem that has not been solved yet is how machine reading tools can be evaluated and compared without available benchmarks or best practices. How can we measure the precision and recall of a method that structures unstructured text or produces formal knowledge? When machine reading tools need to be compared, data-sets are built and annotated according to some guidelines and gold standards are thus created for the underlying domain of application. As an example, authors in [5] show the annotation efforts for an entire year of NewsReader, a European project related to financial and economic data for decision making. They defined the guidelines and several features (such as entity type, factuality, certainty, polarity and time attribute, etc.) without using any formal framework that could help them with the formalization process. The NLP community is also moving towards similar objectives, e.g., with the AMR initiative [6]. AMR implements a simplified, standard neo-Davidsonian semantics using standard feature structure representation where predicates senses and core semantic roles are drawn from the OntoNotes project.[2]

* Corresponding author.
E-mail addresses: aldo.gangemi@istc.cnr.it (A. Gangemi), diego.reforgiato@unica.it (D.R. Recupero), misael.mongiovi@istc.cnr.it (M. Mongiovì), andrea.nuzzolese@istc.cnr.it (A.G. Nuzzolese), valentina.presutti@istc.cnr.it (V. Presutti).

[1] *Abstractive* means that the result of text analysis is not a (set of) text segment(s), but rather a *representation* of a text in a knowledge representation language, cf. [4] for a definition of *abstractive* techniques in NLP.
[2] https://catalog.ldc.upenn.edu/LDC2013T19.

http://dx.doi.org/10.1016/j.knosys.2016.05.023
0950-7051/

Which formal semantics should be employed when reusing machine reading output is not yet agreed. For example, knowledge extraction for the SW is mostly evaluated based on NLP benchmarks (cf. the discussion in [7] and the recent work in [8]). Although they provide solutions for a wide set of SW methods, there are still many metrics, quality measures and problems left uncovered. Moreover, there is nothing there that allows an organization of the tree banks in a structural way with respect to OKE and SW tasks. We argue the urgency and opportunity to define OKE tasks and their associated benchmarks, which would provide the SW community with a native platform to assess research advancement. We think that the right direction is creating *RDF graph banks*, which would radically change OKE research similarly to how syntactic tree-banks [9] did in computational linguistics. A tree-bank is a text corpus annotated with a syntactic sentence structure, providing large-scale empirical data for NLP tasks evaluation. An RDF graph bank is a text corpus annotated with an RDF graph structure. Extending the approach of treebanks, Ontonotes [10], Groeningen Meaning Bank [11], and the semantic banks expressed in Abstract Meaning Representation [6], RDF graph banks can be validated by experts in both SW and linguistics, and can be used as benchmarks for evaluating tools, for learning machine reading models, and to design new tools.

In this paper we identify RDF "motifs" that are as close as possible to good practices in SW and LD. Some elementary patterns (motifs) are defined in order to partition any graph bank into semantically homogeneous subsets. Such motifs correspond to typical logical patterns used in SW and LD. Then we build two sample RDF graph banks (extracted from 100 text sentences and 151 text sentences), and show how they can be validated and refined by RDF experts.

The paper is organized as follows. Section 2 presents the background context of the problem. In Section 3 we describe FRED, the machine reader we have developed and whose graphs we have used as sources for motif identification and for the production of the sample RDF graph bank. In Section 4 we formally define motifs. In Section 5 we present a list of relevant motifs and show how to identify them. Section 6 describes two examples of graph banks created by using motifs, and show how it can be used to evaluate SW tasks. Section 7 shows how we have extended, derived and identified the identified motifs to create a successful application [12,13] of Semantic Sentiment Analysis showing the sentiment motifs. Section 8 ends the paper with conclusions, challenges and possible directions for SW machine reading and graph banks.

2. Background

NLP and SW. The integration between Natural Language Processing (NLP) and SW, under the hat of "semantic technologies", is progressing fast. Most work has been opportunistic: on the one hand exploiting NLP algorithms and applications (typically named-entity recognizers and sense taggers) to populate SW data-sets or ontologies, or for creating NL query interfaces; on the other hand exploiting large SW data-sets and ontologies (e.g., DBpedia,[3] YAGO,[4] Freebase,[5] etc.) to improve NLP algorithms. For example, large text analytics and NLP projects such as Open Information Extraction (OIE, [14]), Alchemy API,[6] and Never Ending Language Learning (NELL, [15]), perform grounding of extracted named entities in publicly available identities such as Wikipedia, DBpedia and Freebase. The links between the two areas are becoming tighter, and clearer practices are evidently needed. Standardization attempts have been introduced with reference to linguistic resources (WordNet,[7] FrameNet,[8] and the growing linguistic linked open data cloud), and the recent proposal of Ontolex-Lemon by the Ontolex W3C Community Group[9] will possibly improve resource reuse. Recently, platforms such as Apache Stanbol,[10] NIF [16] and the NLP2RDF project [17], NERD [18], FOX,[11] FRED [19][12] made it simpler to reuse NLP components as LD, as well as to evaluate them on reference benchmarks, as with GERBIL [8].[13]

Semantic interoperability issues. Interoperability efforts so far mainly focused on the direct transformation of NLP data models into RDF. When this is apparently simple, as in named entity resolution (a.k.a., entity linking), semantic interoperability problems are not so evident. On the contrary, with more advanced tasks, such as relation extraction, compositional analysis of terms, taxonomy induction, frame detection, etc., those problems become evident, and when different results should be combined in order to form a formally and pragmatically reliable ontology, advanced solutions are needed. In practice, even in entity linking, semantics is not as trivial as expected, and explicit assumptions have to be taken about what it means to represent e.g., both `dbpedia:Barack_Obama` and `dbpedia:African_American` as OWL individuals, or to create an `owl:sameAs` link between two resources. Classical work on ontology learning such as [20] takes the integration problem from a formal viewpoint, and uses linguistic features to extract occurrences of logical axioms, such as subclass of, disjointness, etc. Some work from NLP followed a similar direction [21], e.g., NELL relation properties and ontology [22], and "formal semantics" applied to NL (e.g., [23], [24]). These works assume some axiomatic forms, and make the extraction process converge to that form. This is good in principle, but the current state of the art does not really help with establishing clear cut criteria on how to convert NL extractions to RDF or OWL.

From the perspective of NLP, there are a (few) approaches from natural language formal semantics which output formal data structures, but they are not easily interpretable into SW languages. For example, Discourse Representation Structure (DRS), as shown in the output of Boxer [23], is a first-order logic data structure that heavily uses *discourse referents* as variables to anchor the predicates into extensional interpretations, and a *boxing* representation that contextualises the scope of logical (boolean, modal, inferential) operators. Both issues need non-trivial decisions on the side of RDF and OWL design, such as (i) what variables should be accommodated in a SW representation, or ignored? (ii) What logical operators can be safely represented in the formal semantics supported by SW languages? (iii) What predicates should be represented, and in which form, in RDF or OWL?

From the perspective of LD, even porting the original NLP tools data structures into RDF can be beneficial (cf. e.g., the LODifier method [25]), but the reuse of those data will require some intelligence to be integrated. Our stance is that LD are better served if NLP results are reused with a shared semantics that is ready to be integrated with existing RDF data. For example, if a NLP tool outputs data about *Barack Obama* (i.e., its roles, types, relations to other entities), we should be ready to integrate those data to e.g., http://dbpedia.org/resource/Barack_Obama , so that the integrated data preserve their semantics, modulo updates or

[3] http://wiki.dbpedia.org/.
[4] http://datahub.io/dataset/yago.
[5] https://www.freebase.com/.
[6] http://www.alchemyapi.com.
[7] https://wordnet.princeton.edu/.
[8] https://framenet.icsi.berkeley.edu/fndrupal/.
[9] http://www.w3.org/community/ontolex/wiki/Main_Page.
[10] http://stanbol.apache.org.
[11] http://aksw.org/Projects/FOX.html.
[12] http://wit.istc.cnr.it/stlab-tools/fred.
[13] http://aksw.org/Projects/GERBIL.html.