Extracting Knowledge from Text with PIKES

Francesco Corcoglioniti, Marco Rospocher, and Alessio Palmero Aprosio

Fondazione Bruno Kessler—IRST, Via Sommarive 18, Trento, I-38123, Italy
{corcoglio,rospocher,aprosio}@fbk.eu

Abstract. In this demonstration we showcase PIKES, a Semantic Role Labeling (SRL)-powered approach for Knowledge Extraction. PIKES implements a rule-based strategy that reinterprets SRL output in light of other linguistic analyses, such as dependency parsing and co-reference resolution, thus properly capturing and formalizing in RDF important linguistic aspects such as argument nominalization, frame-frame relations, and group entities.

1 Overview

PIKES[1] is a tool for extracting knowledge from natural language text. By exploiting several state-of-the-art Natural Language Processing (NLP) tools, PIKES identifies entities and (complex) entity relations in an English text, and exposes the extracted content in RDF according to Semantic Web and Linked Data best practices.

PIKES builds on Semantic Role Labeling (SRL) for deep text analysis. SRL tools identify occurrences of *frames*, i.e., prototypical situations, in a text, marking the spans of text acting as predicate and role fillers (e.g., 'resigned' and 'the president' in 'the president resigned') and disambiguating them with respect to frame catalogs such as PropBank, NomBank, FrameNet or VerbNet. Differently from other SRL-based approaches, that mainly encode the SRL output in RDF triples, we implemented a rule-based strategy that interprets the information contained in frames and frame-roles occurring in text taking also into consideration other aspects such as the syntactic structure of a sentence, given by the dependency parse tree, and co-reference resolution. This enables to properly capture and represent, among others, situations such as:

argument nominalizations e.g., from 'Joseph Blatter, the president of FIFA [..]', we capture both aspects of 'president',[2] the predicate itself and its implicit subject, thus correctly extracting the information that the implicit president is Joseph Blatter, and he is related to FIFA in a 'president' frame (NomBank `president.01`);[3]

frame-frame relations e.g., from 'Joseph Blatter became president of FIFA in 1998', we relate the information from the frames 'become' and 'president' so that Joseph Blatter, the subject of the 'become' frame, is coreferred to the subject of the 'president' frame, and a relation between the two frames is explicitly represented;[4] similarly, from 'Joseph Blatter ended up resigning from FIFA in 2015', we correlate the

[1] http://pikes.fbk.eu/

[2] In Nombank (http://bit.ly/nombank) for some predicate nouns, the predicate can be its own argument (usually ARG0). Examples include: teacher, president, director.

[3] The complete output for this example and its graphical representation obtained with the on-line demo of the tool (see Section 3) are directly accessible from here: http://bit.ly/pikes-argnorm.

[4] See http://bit.ly/pikes-frmfrm.

information from the frames ‘end’ and ‘resign’ so that Joseph Blatter, the subject of the ‘end’ frame, is coreferred to the ‘president’ argument of the ‘resign’ frame, and a relation between ‘end’ and ‘resign’ is represented;[5] and,

group entities e.g., given a sentence like ‘Joseph Blatter and João Havelange led FIFA [..]’, we extract that two distinct entities, Joseph Blatter and João Havelange, are both subject of the frame ‘lead’ (while SRL tools typically annotate the whole span of text ‘Joseph Blatter and João Havelange’ as a single argument).[6]

Besides better interpreting the SRL information, our approach adopts a representation model where all the content processed and produced in extracting knowledge – the textual input and its metadata (e.g., author, date/time), the intermediate output produced by NLP tools (e.g., extracted SRL frames), the final output of the system – is organized in three interlinked layers – Text, Mentions, Instances – and exposed as RDF according to Semantic Web best practices. Furthermore, by exploiting linking between layers, we relate each triple extracted from text to the span(s) of text and the intermediate NLP output from where it was derived, thus enabling a fine-grain tracking of the whole extracted content that helps debugging and improving the knowledge extraction process.

2 Under the hood

PIKES works in two main phases that we briefly describe below; more details on the NLP tools used and the knowledge extraction algorithm are reported on the website.

In the first phase (from Text to Mentions), the input text is processed by several NLP tools to extract *mentions*, i.e., pieces of text denoting something of interest, such as an entity or relation. Mentions represent, in a structured form, all the information needed to extract the knowledge conveyed by the text. For instance, given the example text[7]

> *G.W. Bush and Bono are very strong supporters of the fight of HIV in Africa.*
> *Their March 2002 meeting resulted in a 5 billion dollar aid.*

the span of text “G.W. Bush” corresponds to a mention that is identified by a URI (e.g., `<resource_uri#char=0,10>`) and has several attributes such as a textual extent (‘G.W. Bush’), a position in the text (characters 0 to 10), a type (e.g., `NameMention`), a possible corresponding DBpedia entity (e.g., `dbpedia:George_W._Bush`), and so on. All these attributes, together with the input text metadata (e.g., author, creation time), are also exposed by PIKES as RDF. PIKES currently relies on state-of-the-art NLP tools such as: *Stanford CoreNLP*[8] for part-of-speech tagging, named entity recognition and classification, temporal expression recognition and normalization, and coreference resolution; *mate-tools*[9] for dependency parsing and SRL; *DBpedia Spotlight*[10] for entity linking; and, *UKB*[11] for word sense disambiguation (with respect to WordNet 3.0). Furthermore, we developed a dedicated module for mapping the NLP annotations produced by all these tools to mentions and mentions attributes expressed in RDF.

[5] See http://bit.ly/pikes-frmfrm2.

[6] See http://bit.ly/pikes-group.

[7] Try it on PIKES: (http://bit.ly/pikes-example)

[8] http://nlp.stanford.edu/software/corenlp.shtml

[9] https://code.google.com/p/mate-tools/

[10] http://spotlight.dbpedia.org/

[11] http://ixa2.si.ehu.es/ukb/

In the second phase (from Mentions to Instances), mentions are processed with mapping rules that match certain mention attributes/patterns and create consequent RDF triples. Mapping rules are formulated as SPARQL Update **INSERT**… **WHERE**… statements that are repeatedly executed until a fixed-point is reached. Rules are allowed to create new individuals, can invoke external code by means of custom SPARQL functions and can access and match also data in auxiliary resources (e.g., for mapping purposes) as well as the instance data created so far. Current rules can be organized in six categories based on their function: (i) for creating new instances; (ii) for typing extracted instances (i.e., generate `rdf:type` assertions), based on mention attributes (e.g., WordNet synset, PropBank/NomBank roleset, NERC class); (iii) for adding annotations (e.g., `rdfs:label` and `foaf:name` assertions) from the textual extent of the mention; (iv) for linking (via `owl:sameAs` or `rdfs:seeAlso` assertions) an instance and the corresponding DBpedia resource; (v) for relating (with the proper properties) frame instances to argument instances; and, (vi) for linking (via `owl:sameAs` assertions) the instances corresponding to coreferential mentions. The resulting RDF is also post-processed materializing implicit knowledge and discarding unnecessary data. All the processing in this second phase is performed exploiting RDF$_{\text{pro}}$ [1], an RDF manipulation tool which we extended with additional plugins for SPARQL-like rule execution and named-graph normalization. For instance, from the mention we previously considered (`<resource_uri#char=0,10>`), several triples are instantiated, e.g.,

```
dbpedia:George_W._Bush rdfs:label ``G. W. Bush'' ;
                       foaf:name ``G. W. Bush'' ;
                       rdf:type dbyago:HeadOfState110164747 ;
                       rdf:type sumo:Entity ; ...
```

The input text, its mentions, and the triples extracted are related by various properties that enable to state that a mention *is part of* an input text, or that a mention *expresses* a triple (i.e., the triple can be derived from the mention). In the latter example, instead of reifying the assertion, we prefer to use named graphs and keep the RDF representation more compact. In particular, each extracted triple is placed in a named graph that represents the set of mentions (in some cases a single mention) that express that particular triple. Clearly, a named graph may contain many triples, meaning that all these triples were extracted from the same mention (e.g., different type assertions on the same instance), and a named graph may be expressed by many mentions, meaning that all the triples in the named graph were extracted from each one of these mentions.

3 PIKES in action

PIKES is publicly accessible through an on-line demo version,[12] where users can freely test our approach on sentences of their choice. To run the demo, users just have to type in a text, and press the submit button. Several tabs become available once PIKES finishes processing the input text: one example, the graphical rendering of the knowledge extracted by the tool, is shown in Figure 1. In particular: the *Metadata* tab reports the RDF encoding of the metadata attached to the input document, as well as a summary of the modules applied to extract knowledge; the *Mentions* tab reports the RDF serialization of the mentions identified in the input text by NLP tools, together with

[12] Accessible from http://pikes.fbk.eu. An explanatory demo video is also available.

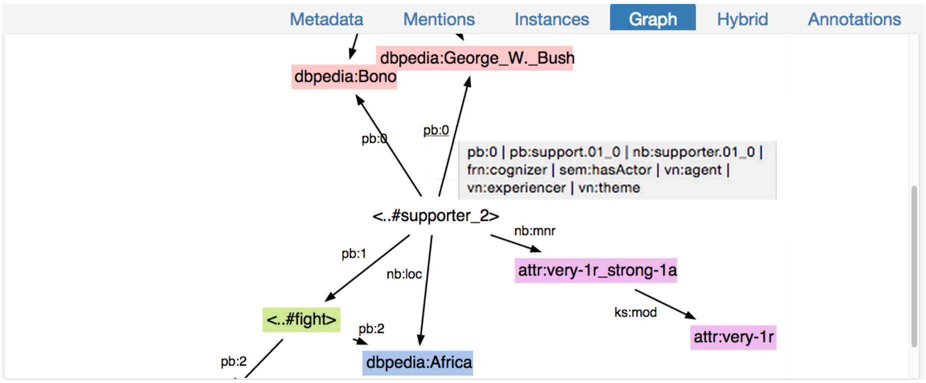

Fig. 1. Graphical rendering (excerpt) of the knowledge extracted from the example text in Sec. 2.

their attributes;[13] the *Instances* tab shows the content of the Instances layer, i.e., the actual triples distilled from the mention information, representing the final output of the system; these triples are also graphically rendered in the *Graph* tab, where nodes represent instances, and arcs assertions on them; additional assertions (e.g., types, labels, etc) are shown by tooltips when hovering with the mouse over any element of the graph (e.g., the grey box in Figure 1); finally, the *Hybrid* tab highlights, sentence by sentence, the mentions where they occur in the text (hovering over the annotations, mention attributes can be accessed) as well as the graph of the corresponding instances extracted from each single sentence (by clicking on the sentence ID).

4 Concluding Remarks

PIKES was applied to extract knowledge from the whole Simple English Wikipedia,[14] consisting of ~110K text documents long 219 words on average. The processing took 507 core hours (16 s per wikipedia page on average), and was completed by 16 parallel instances of PIKES in less than 32 hours. The resulting RDF dataset (including textual metadata, mentions and mention attributes, extracted triples) is available for download.[15] We also performed an evaluation of the output,[16] obtaining an average precision of 85.4% on a random sample of 200 triples (2 annotators, Fleiss's kappa 0.372), which demonstrates how PIKES can efficiently extract accurate knowledge from text.

Acknowledgements The research leading to this paper was supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

References

1. Corcoglioniti, F., Rospocher, M., Mostarda, M., Amadori, M.: Processing Billions of RDF Triples on a Single Machine using Streaming and Sorting. In: ACM SAC 2015 Proceedings

[13] The *Annotations* tab shows the raw annotation file produced by NLP tools used in PIKES.
[14] http://simple.wikipedia.org/
[15] http://pikes.fbk.eu/sew-rdf.html
[16] http://pikes.fbk.eu/evaluation.html