From Papers to Triples: An Open Source Workflow for Semantic Publishing Experiments

Bahar Sateli and René Witte

Semantic Software Lab, Department of Computer Science
and Software Engineering, Concordia University, Montréal, Canada

Abstract. In this demonstration paper, we describe an open source workflow for supporting experiments in semantic publishing research. Based on a flexible, component-based approach, natural language papers can be converted into a Linked Open Data (LOD) compliant knowledge base. We exemplary discuss how to plan and execute experiments based on an integrated suite of tools, thereby both significantly lowering the barrier of entry in this field, while also encouraging the exchange of tools when building novel contributions.

1 Introduction

Semantic publishing research aims at making scientific publications readable and semantically understandable to computers. The long-term vision is to enable automated agents supporting researchers in their daily tasks: Finding literature pertaining to a tasks, automatically summarizing state-of-the-art, connecting experiments and datasets, or finding novel contributions in a field. A number of approaches in this area build on the standards and tools from the Semantic Web initiative [1], such as the Resource Description Framework (RDF) and its vocabularies RDF Schema (RDFS), which is well-supported by numerous open source tools.

When dealing with the huge amount of existing literature, a required foundation for performing experiments in this area is an automated, robust process for converting natural language texts into a Linked Open Data (LOD) [4] compliant format. The resulting knowledge base can then be easily inter-linked with other (research) entities on the web of data, supporting numerous use cases in semantic publishing research. Here, we describe a workflow and its implementation, based on a combination of our own with existing open source infrastructure, including natural language processing (NLP) components, entity grounding to the LOD cloud, and converting NLP results to RDF triples. This approach has been successfully applied in a number of semantic publishing experiments, including building semantic wiki user interfaces for literature management [6], semantic literature querying [8], the Semantic Publishing Challenge (2015) Task 2 [7] and semantic profiling of scientists based on their publications.

2 Converting Papers to Triples: Processing Workflow

The approach demoed here has the core assumption that all relevant information for building semantic publishing applications is extracted from textual artifacts and inserted

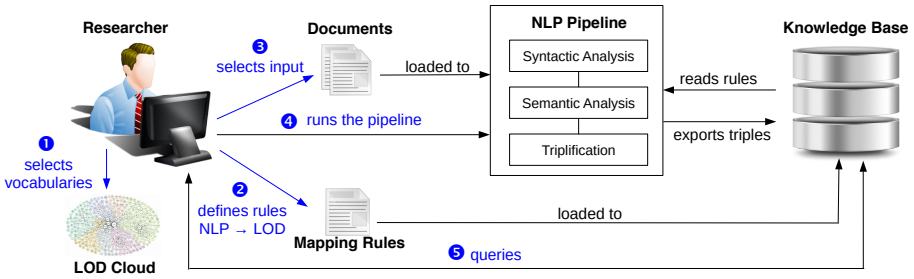


Fig. 1: Workflow for triplification of scientific literature

into a LOD-compliant knowledge base (KB). We start from a set of documents, typically scientific articles (e.g., conference papers or journal articles), but possibly also other texts, like dataset descriptions or scientific tool documentations. For the scope of this demonstration, we exclude a discussion on text extraction from PDF documents; a comprehensive overview can be found in [2]. By default, the GATE text analysis framework [3] used here relies on the Apache Tika library[1] for converting different file types, such as Word, ODT, PPT, or PDF, to plain text.

Semantic analysis tools running on the textual documents provide structured descriptions, for example, on entities, citations, rhetorical entities, or writing styles. To make these results available in a form of a knowledge base, we show how we can directly connect the GATE framework with a triplestore using a novel approach we developed based on Apache Jena,[2] the *LODeXporter*. The resulting KB can then be queried for further scientific experiments or leveraged within a user interface, as illustrated in Fig. 1.

2.1 Text Mining Components

The natural language analysis part of our workflow is implemented based on the GATE *(General Architecture for Text Engineering)* framework [3], which provides us with a robust and widely used NLP infrastructure. GATE is designed as a component-based architecture, where individual analysis components (called *processing resources* or PRs) can be easily added, modified, or removed from the system. A document is processed by a sequential *pipeline* of PRs: Each component can read and add results to a text in form of *annotations*, which form a graph over the document. GATE is licensed under the GNU LGPL and can be obtained from http://gate.ac.uk; GATE Embedded libraries are also available through Maven Central. Most of the plugins described below can be easily added through GATE's *Plugin Manager* from our *Semantic Software Lab* repository. Snapshots of our presented code are additionally available on our GitHub repository at https://github.com/SemanticSoftwareLab. Continuous integration services are provided through a Jenkins server at http://assistant.semanticsoftware.info/.

Preprocessing. For preprocessing documents, we rely on the standard components shipped with the GATE distribution, in particular the ANNIE plugin [3]. These perform standard NLP tasks that later steps build upon, such as tokenization, sentence splitting, part-of-speech (POS) tagging, stemming, and verb group analysis.

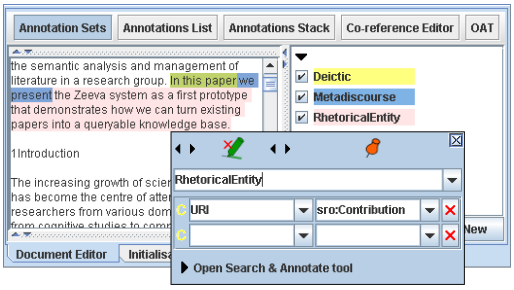
[1] Apache Tika, https://tika.apache.org/
[2] Apache Jena, https://jena.apache.org/

```
Rule: INDeictic (
   {Token.category == "IN", Token.orth == "upperInitial"}
   {Token.category == "DT"}
   {Lookup.majorType ==  "DEICTIC"}
):mention -->
:mention.Deictic = {content = :mention@string}

Rule: ContributionActionTrigger (
   {Deictic} {Token.category == "PRP"}
   ({Token.category == "RB"})?
   {Lookup.majorType == "ACTION"}
):mention -->
:mention.Metadiscourse = {type = "sro:Contribution"}

Rule: RESentence (
   {Sentence, Sentence contains ({Metadiscourse}):meta}
):mention -->
:mention.RhetoricalEntity = {URI = :meta.type}
```

(a) Example JAPE rules



(b) Detected RE annotation in GATE Developer

Fig. 2: JAPE rules (left) to extract a Contribution sentence and the generated annotations, color-coded in GATE's GUI

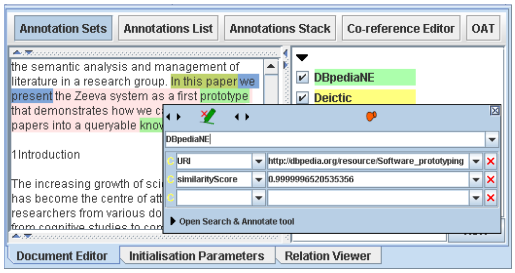
Rhetector. A distinguishing feature of scientific literature, compared to other textual documents, is that sections of a scholarly document usually follow a specific argumentative order. In fact, several de-facto standards exist, such as IMRAD [9], to capture the authors' rhetoric in various domains, with the aim of making scientific communication efficient and organized. A challenging task in the process of text mining scientific literature is to automatically detect these rhetoric zones in a document. The automatic semantic annotation of Rhetorical Entities (REs), such as *contributions* or *claims*, has proven to be effective in finding information on a more granular level, for example, in finding all papers that use a specific method *M* in their experiments [8]. Here, we demonstrate our *Rhetector* component[3] to automatically detect REs in scientific literature, currently limited to Claims and Contributions. For each detected RE, an annotation of type "RhetoricalEntity" is added to the document. Based on the grammatical structure of the detected RE, it is then classified and mapped onto existing concepts on the Linked Open Data (LOD) cloud. Fig. 2 shows a number of hand-crafted rules written in GATE's JAPE language [3] that incrementally detect *deictic* and *meta-discourse* entities in text and classify the encompassing sentence, based on the discourse actions, as a new rhetorical entity. Our Rhetector PR is licensed under the GNU LGPL v3 and available through GATE's Plugin Manager.

LODtagger. To detect domain-specific entities in research publications, we rely on the LOD cloud, in particular DBpedia. This provides for a rich, continuously updated resource in a standard semantic format. By linking entities detected in documents to LOD URIs (Universal Resource Identifiers), we can semantically query a knowledge base for all papers on a specific topic (URI), even when that topic is not mentioned literally in a text: E.g., we can find a paper for the topic "*linked open data*," even when it only mentions "*LOD*," since they are semantically related in the DBpedia ontology. For the actual entity tagging, we rely on an external tool, DBpedia Spotlight [5]. To integrate this web service into a GATE text mining pipeline, we developed *LODtagger*.[4] This component sends the entire UTF-8 formatted text of a document as a RESTful

[3] Rhetector, http://www.semanticsoftware.info/rhetector
[4] LODtagger, http://www.semanticsoftware.info/lodtagger

```
{
  "Resources":
  [{
      "@URI": "http://dbpedia.org/resource/Software_prototyping",
      "@support": "3235",
      "@types": "",
      "@surfaceForm": "prototype",
      "@offset": "1103",
      "@similarityScore": "0.9999996520535356",
      "@percentageOfSecondRank": "0.0015909752111777534"
  }]
}
```


(a) Excerpt of Spotlight JSON response



(b) Generated NE annotation in GATE

Fig. 3: Example response from Spotlight (left) and the generated annotation (right)

POST request to a Spotlight endpoint and receives the results in JSON format, which are subsequently parsed and transformed to GATE annotations (Fig. 3). To further limit the annotations to named entities, we also demonstrate how we can filter LODtagger results using our syntactic MuNPEx[5] noun phrase (NP) chunker, thereby significantly reducing false positives. Our LODtagger PR is also licensed under the GNU LGPL v3 and can be installed through GATE's Plugin Manager.

LODeXporter. As explained above, we aim to create a semantic knowledge base that contains the information extracted from research documents, to be able to inter-link it with other triples in the same KB or the LOD cloud, ultimately supporting end-user applications or data analysis experiments. However, so far we only generated GATE-specific document annotations. Rather than 'hard-coding' a specific export strategy of GATE annotations to triples, we wanted a more flexible solution that can *(a)* be easily extended for new NLP annotations (e.g., when importing new PRs into a pipeline) and *(b)* provide for flexible mapping of annotations to LOD vocabularies, in order to facilitate experiments with different ontologies. Our solution is a novel process where we derive the export process, including the mapping of NLP entities to LOD vocabularies, from a knowledge base. Our *LODeXporter* component directly connects a GATE pipeline to a (currently Jena TDB-based) triplestore. The KB contains rules, expressed in RDF, which describe how a specific GATE annotation should be mapped to triples, as well as the vocabularies to use, such as the one shown below that describes the mapping of a GATE "DBpediaNE" annotation to an RDF triple of type "pubo:LinkedNamedEntity":

```
@prefix map: <http://semanticsoftware.info/mapping/mapping#> .
@prefix pubo: <http://lod.semanticsoftware.info/pubo/pubo#> .

map:GATEDBpediaNE a map:Mapping ;
        map:GATEtype     "DBpediaNE" ;
        map:type         pubo:LinkedNamedEntity ;
```

This way, the same pipeline can support different RDF export results, simply by virtue of changing the triples in the KB or connecting to a different KB. LODeXporter[6] is currently considered to be in pre-release, as we plan to improve the expressiveness of mapping relations between entities before the 1.0 release.

[5] MuNPEx, http://www.semanticsoftware.info/munpex
[6] LODeXporter, http://www.semanticsoftware.info/lodexporter

2.2 Application

At this point, we have a knowledge base populated with the information extracted from research publications. We can now leverage this KB for experiments, by querying it for specific information using SPARQL, e.g., by deploying an Apache Fuseki[7] server. The general challenge is to formulate a scientific hypothesis that can be empirically evaluated based on the query results. In this demo, we show how we evaluated a number of concrete tasks, such as querying documents based on entities alone vs. entities appearing in rhetorical zones [8] or matching documents to semantic user profiles. While some experiments can be performed solely based on the generated triples, others will require a user evaluation or comparison to a gold standard. However, we cannot cover these steps within the scope of this demonstration paper.

Of course, the generated KB can also be used to drive end-user applications for evaluating the impact of semantic support on concrete scholarly tasks (e.g., literature review) – for example, through a semantic wiki system [6].

3 Conclusions

We described a workflow for experiments in semantic publishing research that is based entirely on open source tools and standards. The demo will highlight how this process can be easily configured for different research questions. We hope that the presented work will help others to adapt semantic text mining tools in their projects.

References

1. Berners-Lee, T., Hendler, J.: Publishing on the semantic web. Nature 410(6832) (2001)
2. Constantin, A., Pettifer, S., Voronkov, A.: PDFX: Fully-automated PDF-to-XML Conversion of Scientific Literature. In: Proceedings of the 2013 ACM Symposium on Document Engineering. pp. 177–180. DocEng '13, ACM, New York, NY, USA (2013)
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011), http://tinyurl.com/gatebook
4. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis lectures on the semantic web: theory and technology, Morgan & Claypool Publishers (2011)
5. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proc. 7th Intl. Conf. on Semantic Systems. pp. 1–8. ACM (2011)
6. Sateli, B., Witte, R.: Supporting Researchers with a Semantic Literature Management Wiki. In: The 4th Workshop on Semantic Publishing (SePublica 2014). CEUR Workshop Proceedings, vol. 1155. Anissaras, Crete, Greece (May 25 2014), http://ceur-ws.org/Vol-1155/paper-03.pdf
7. Sateli, B., Witte, R.: Automatic Construction of a Semantic Knowledge Base from CEUR Workshop Proceedings. In: Semantic Web Evaluation Challenges: SemWebEval 2015 at ESWC 2015 , Portorož, Slovenia, May 31 – June 4, 2015, Revised Selected Papers. Communications in Computer and Information Science, vol. 548, p. 129–141. Springer (2015)
8. Sateli, B., Witte, R.: Semantic representation of scientific literature: bringing claims, contributions and named entities onto the Linked Open Data cloud. PeerJ Computer Science 1(e37) (2015), https://peerj.com/articles/cs-37/
9. Sollaci, L.B., Pereira, M.G.: The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. Journal of the medical library association 92(3), 364 (2004)

[7] Apache Fuseki, https://jena.apache.org/documentation/serving_data/