Automatic Construction of a Semantic Knowledge Base from CEUR Workshop Proceedings

Bahar Sateli and René Witte(✉)

Semantic Software Lab, Department of Computer Science
and Software Engineering, Concordia University, Montréal, Canada
witte@semanticsoftware.info

Abstract. We present an automatic workflow that performs text segmentation and entity extraction from scientific literature to primarily address Task 2 of the Semantic Publishing Challenge 2015. The goal of Task 2 is to extract various information from full-text papers to represent the context in which a document is written, such as the affiliation of its authors and the corresponding funding bodies. Our proposed solution is composed of two subsystems: *(i)* A text mining pipeline, developed based on the GATE framework, which extracts structural and semantic entities, such as authors' information and references, and produces semantic (typed) annotations; and *(ii)* a flexible exporting module, the LODeXporter, which translates the document annotations into RDF triples according to custom mapping rules. Additionally, we leverage existing Named Entity Recognition (NER) tools to extract named entities from text and ground them to their corresponding resources on the Linked Open Data cloud, thus, briefly covering Task 3 objectives, which involves linking of detected entities to resources in existing open datasets. The output of our system is an RDF graph stored in a scalable TDB-based storage with a public SPARQL endpoint for the task's queries.

1 Introduction

Semantic Publishing is a new, thriving research domain, driven by a synergic community of semantic web researchers, computational linguists, librarians and publishing companies, all aiming towards a platform for the dissemination of scientific literature, accessible to both humans and machines. The vision is to develop tools and frameworks to enrich scholarly literature with metadata in order to facilitate retrieval, automatically exploiting and evaluating research artifacts, such as articles and datasets. The ever-increasing amount of available scientific literature, however, has rendered manual efforts of annotating documents ineffective. Consequently, researchers are in dire need of automatic systems that can detect various entities from scientific literature and make them available in open formats.

The *Semantic Publishing Challenge*, started in 2014, is a recent series of competitive efforts to produce linked open datasets from multi-format and multi-source

F. Gandon et al. (Eds.): SemWebEval 2015, CCIS 548, pp. 129–141, 2015.
DOI: 10.1007/978-3-319-25518-7_11

input documents. The 2015 edition of the challenge[1] targeted the automatic analysis of several computer science workshop proceedings to extract fine-grained bibliographical metadata from workshops' full-text papers. The dataset under study is composed of 183 workshop papers, published between 2007 and 2014 by CEUR-WS.org. The challenge is to automatically extract authors, affiliations, cited works, funding bodies and mentioned ontology names from the text and populate a knowledge base, in which all the detected entities are semantically described and inter-linked with each other, where applicable.

The generated knowledge base is finally evaluated against a set of 10 predefined queries for its correctness and completeness and exploited as a means of assessing the quality of scientific production in the respective workshops. The challenge queries are concerned with searching for entities, categorized as follows:

- Authors, their Affiliations (**Q2.1**) and the country where the affiliation is located in (**Q2.2**);
- References cited in a paper (**Q2.3**), their year of publication (**Q2.4**), and type (**Q2.5**);
- Research Grant numbers (**Q2.6**), names of Funding Agencies (**Q2.7**) and European Projects (**Q2.8**) supporting the research presented in the paper; and
- Names of existing (**Q2.9**) and new (**Q2.10**) Ontologies mentioned in a paper.

In this paper, we present our automatic workflow that performs text segmentation and entity detection to address Task 2 of the challenge. Our system is able to extract contextual information, such as the entities required to answer the challenge queries, from the full-text of the given papers, and make them available as a linked open dataset. Additionally, we briefly cover Task 3 objectives, by linking named entities that appear in the documents to their corresponding resources on the Linked Open Data (LOD) cloud, whenever possible. We leverage a combination of multiple techniques from the Natural Language Processing (NLP) and Semantic Web domains to automatically construct a semantic representation of the knowledge contained in a scientific document. We believe that such a rich representation can pave the way for a variety of advanced use cases, such as creating automatic literature reviews, facilitating information synthesis and literature-based knowledge discovery. Note that you can find supplementary material, such as the populated knowledge base and the text mining pipeline resources at http://www.semanticsoftware.info/sempub-challenge-2015.

2 Design

The ultimate goal of our approach is to automatically extract the entities needed to answer the challenge queries from the given dataset and store them in a knowledge base with semantic metadata. In our approach, we use text mining to detect the desired entities from a document's full-text. Given the lack of training

[1] Semantic Publishing Challenge 2015, https://github.com/ceurws/lod/wiki/SemPub2015.

data for computer science literature, we decided to adopt a rule-based approach, as opposed to applying machine-learning techniques.

Figure 1 provides a high-level overview of our system. The NLP pipeline accepts a document as input, which goes through multiple processing phases, and produces semantic triples as output. The *Syntactic Processing* phase breaks down full-text of the document into smaller segments and pre-processes the text for further semantic analysis. The *Semantic Processing* phase takes the results of syntactic analysis and attempts to annotate various entities in text. Finally, the document's annotations will be translated into semantic triples according to a series of custom *mapping rules* and made persistent in a knowledge base. Throughout this section, we provide examples from the challenge training dataset to clarify our approach. Each example sentence will also bear a reference to its corresponding paper.

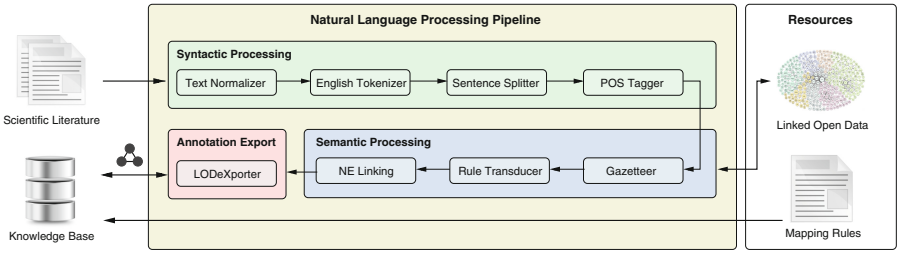


Fig. 1. Automatic workflow to transform scientific literature into a knowledge base

2.1 Syntactic Processing

The input of our text mining pipeline are documents (e.g., the dataset PDF files) containing the collected work of its authors in a descriptive format, as well as other additional content, like title, figures and references. In our pipeline, we first scrape the text of documents and normalize the output, such as, whitespace trimming and faulty character encoding replacement. As a prerequisite step, we then break down the content of the document into individual tokens,[2] sequences of tokens (e.g., n-grams) and sentences. Since our semantic processing components rely on specific characteristics of sentences, like their verbs, we also label each lexical item in a sentence with a Part-Of-Speech (POS) tag, like *adjective* or *pronoun*, as its grammatical category. The pre-processed text is subsequently passed onto the semantic processing subsystem for entity detection.

2.2 Semantic Processing

The semantic processing subsystem is responsible for detecting entities required for the challenge queries from text and generate typed annotations as output. Here, we provide a detailed description of each of our subsystem's components.

[2] Tokens are smallest, meaningful units of text, such as words, numbers or symbols.

Gazetteer. The *Gazetteer* component is essentially a dictionary with several lists of carefully curated words that are matched against the text to mark tokens for further processing. In addition to reusing GATE's gazetteer of person and location names for author and affiliation extraction, we curated a list for detection of segment headers (7 entries), as well as a list of general terms used in computer science (30 entries), discourse deictic cliches (8 entries), and verbs used in the scientific argumentation context (160 entries) for rhetorical analysis of documents. We curated these gazetteer lists – a subset of which is shown in Table 1 – from manual inspection of the training dataset documents and Teufel's AZ corpus[3] for rhetorical entities. The role of the Gazetteer component is to compare the text tokens against its dictionary entries and generate so-called *lookup* words subsequently utilized within our entity detection rules.

Table 1. A subset of our text mining pipeline's gazetteer lists

List Type	Example Entries
Domain Concepts	*framework, algorithm, approach, position paper, article*
Rhetorical Verbs	PRESENTATION: *describe, present, put forward, demonstrate* SOLUTION: *propose, overcome, address, enhance, achieve* ACTION: *investigate, apply, assess, develop, construct*
Deictic Cliches	*to deal with this problem, towards this end, in what follows*
Segment Headers	*Acknowledgments, Keywords, References*
Person First Names	*Richard, Jamshaid, Olaf*
Organization Prefixes	*Federal, National, Freie, Open*
Organization Base	*University, Universität, Department, Faculty, Institute*

Rule Transducers. The *Rule Transducers* are responsible for detecting the desired entities of the challenge. Transducers apply pattern-matching rules to classify the text tokens and sentences into one of several pre-defined classes (or none). The input to the transducers are sentences, word tokens with their POS and root form,[4] as well as the lookup words marked by the Gazetteer component. Whenever a match is found in text, this component annotates the boundary of the matched sequence with a semantic type, such as Author or Title. We developed several rules for the following categories:

Text Segmentation. Based on segment headers detected by the Gazetteer component, we blindly annotate the span between each two headers (and Start-of-Document and End-of-Document) with the corresponding header as its class. For example, we annotate everything from the start of the document until the word "*Abstract*" as the document's Metadata_body.

[3] Argumentation Zoning (AZ) Corpus, http://www.cl.cam.ac.uk/~sht25/AZ_corpus.html.

[4] The root or *lemma* of a word is its canonical form without any inflectional endings.

Authors. The person name detection is based on the tokens marked by the Gazetteer component as first names. All first name tokens followed by an upper initial token are annotated as Persons in text. Subsequently, we extract each Person name in the document's Metadata_body (excluding the ones that appear within an organization name) as an Author annotation.

Affiliations. We designed several rules to capture various patterns of organization names, limited to academic institutions, from the document's metadata body. We also capture the geographical location of the organization from *(i)* the name of the institution, or *(ii)* the location name mentioned closest to the organization, in terms of its start offset in text. We retain the detected location name along with the affiliation annotation in order to answer query **Q2.2** of the challenge (see Sect. 1).

(1) *"University of **Trento, Italy**"*$_{(Vol315\_paper01)}$

(2) *"The Open University, **UK**"*$_{(Vol523\_deWaard)}$

Authors-Affiliations Relations. We developed a separate processing resource that implements multiple heuristics to extrapolate which Authors are employed by a detected Affiliation entity. If both Author and Affiliation mentions in text are indexed (e.g., with numbers or symbols), the matching is performed based on the indices. Otherwise, the processing resource merely infers such a relationship between each Author and its closest Affiliation annotation using their start offsets in text. Subsequently, the result of the matching process is stored as the "*employedBy*" feature of the Author annotation.

References. Detection of references titles, authors and publishing venue is one of the most challenging parts of document analysis, mostly due to inconsistencies in bibliographical styles used in the papers (e.g., see Vol-721[5] in the dataset). We tackled this problem by hand-crafting rules for multiple styles, including `abbrv` and `plain` classes used in the training set. We break down the References_body segment into smaller fragments: Similar to author names described above, we detect author names and paper title from each reference. We then annotate the tokens in between the paper title and the year of publication (or End-of-Line) as the publishing venue. References are eventually categorized into either "*journal*" or "*proceedings*" classes based on whether a journal citation (volume, number and pagination) is present, like the ones shown below:

(3) *"G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig.ma: Live views on the web of data. **Journal** of Web Semantics, **8(4):355-364**, 2010"*$_{(ldow2011\_paper10)}$

(4) *"M. Hausenblas, "Exploiting linked data to build applications," IEEE Internet Computing, **vol. 13, no. 4, pp. 68-73**, 2009"*$_{(ldow2011\_paper12)}$

[5] Task 2 Dataset, https://github.com/ceurws/lod/wiki/Task2#data-source.

Ontologies. Ontology name detection is performed using the root form of word tokens. We capture three forms of ontology mentions: *(i)* concatenated or camel-case ontology names, *(ii)* upper initial ontology names, and *(iii)* acronyms or all-caps tokens mentioned in a sentence on a fixed window distance from the word "*ontology*".

(5) *"...two versions of an ontology of the hydrographical domain:* ***hydrOntology****."*$_{(Vol571\_paper4)}$

(6) *"...the* ***Privacy Preference Ontology (PPO)*** *that enables users to create..."*$_{(ldow2011\_paper01)}$

(7) *"The* ***GoodRelations Ontology*** *is experiencing the first stages of ..."*$_{(ldow2011\_paper12)}$

Contributions. An interesting subtask of the challenge is to find the new ontologies introduced in a paper. To this end, we attempt at finding sentences in the document's abstract that describe the Contributions of the authors. We first look for *deictic* phrases, such as "*in this paper*". Deictic phrases are expressions within an utterance that refer to parts of the discourse. For example, the word "*here*" in "*here, we describe a new methodology...*" refers to the article that the user is reading. In scientific literature, deictic phrases are often used in sentences that provide a high-level overview of what is presented in the paper, referred to as the *metadiscourse* elements, such as the following examples:

(8) *"In this paper we introduce the Publishing Workflow Ontology (PWO)..."*$_{(Vol1302\_paper01)}$

We designed hand-crafted rules to capture Contribution sentences that look at sequences of deictic phrases, metadiscourse mentions and the rhetorical function of the verbs mentioned in the sentence [1]. Note that we require an explicit reference to the agent (i.e., authors) or the discourse deixis in each sentence. Subsequently, each sentence containing a metadiscourse element followed by a noun phrase is annotated as a Contribution entity. Finally, the ontologies mentioned in the Abstract section within the boundary of a Contribution are extracted for **Q2.10** of the challenge (see Sect. 1).

Funding Agencies. Funding agency mentions in text are extracted from the Acknowledgement segment of each paper. The agency name is detected as either *(i)* one or more upper-initial word tokens, or *(ii)* an organization name. We plan to integrate a parsing component into our text mining pipeline, so that the funding agency name can be extracted from the noun phrase following the "*funded by*" verb phrase in the sentence's dependency tree.

(9) *"This work has been funded by the* ***SemanticHealthNet Network of Excellence****..."*$_{(Vol1302\_paper06)}$

(10) *"This work was partially supported by* ***European Commission****..."*$_{(Vol1118\_paper1)}$

NE Linking. Previously, we investigated how we can use generic Named Entity Recognition (NER) tools to extract topics from scientific literature in a domain-independent manner, as a means of modeling the knowledge in a paper [1]. In our text mining pipeline, we use external NER components to extract topics (named entities) of the document and link them to their corresponding resources on the LOD cloud [1].

2.3 Knowledge Base Construction

In order to generate a semantic representation of the detected entities described in the previous sections, we export all annotations into semantic triples using the W3C RDF[6] standard to construct a knowledge base. While the type of annotations, e.g. Affiliation, is determined by the Rule Transducers component, we still would like to have the flexibility to express the mapping of annotations to RDF triples and their inter-relations at run-time. This way, various representations of knowledge extracted from documents can be constructed based on the intended use case and customized without affecting the underlying syntactic and semantic processing components.

Reuse of Vocabularies. Conforming to the best practices of producing linked open datasets,[7] we decided to reuse existing open vocabularies to describe both the structural and semantic metadata that we extract from each document. In scientific literature mining, controlled vocabularies are used in form of *markup* languages, which are added to text (either manually or automatically) to annotate various entities of documents.

In order to tolerate the formatting variations of the datasets items (e.g., ACM vs. LNCS, double-column vs. single-column), we decided to remove all formatting from documents during processing and use the DoCO ontology [2] to describe various units of information, such as Sentences or Bibliography section, in the document. DoCO is an OWL 2 DL ontology that serves as a general-purpose vocabulary for describing documents in RDF. Additionally, it integrates DEO[8] and SALT [3] ontologies for annotation of rhetorical entities, such as Contributions, in a scholarly document. By linking to instances of the DoCO ontology, we can attach syntactic and semantic markup to the document, which can be later queried to answer the challenge queries, e.g., by annotating parts of the Abstract text that describe the authors' Contributions, so that we can detect new ontologies introduced in a paper (see **Q2.10** in Sect. 1).

Publication Ontology (PUBO). We developed the *PUBlication Ontology* (PUBO)[9] – a vocabulary for scientific literature constructs that describes a document's various segments (e.g., sentences) and their contained entities. Wherever possible, we reused existing Linked Open Vocabularies (LOV): To express the semantic types of entities, like Sentences and Contributions, we chose to link to DoCO[10] and SALT Rhetorical Ontology (SRO) for our experiments. We also added our own vocabulary to describe the relation between a source document and its contained entities, for example, to describe the topics that appear within the boundary of a rhetorical entity. Our ontology uses "pubo" as its namespace throughout this paper.

[6] Resource Description Framework (RDF), http://www.w3.org/RDF/.
[7] Best Practices for Publishing Linked Data, http://www.w3.org/TR/ld-bp/.
[8] Discourse Elements Ontology (DEO), http://purl.org/spar/deo.
[9] PUBlication Ontology, http://lod.semanticsoftware.info/pubo/pubo.rdf.
[10] Document Components Ontology (DoCO), http://purl.org/spar/doco.

LODeXporter. We designed the *LODeXporter*[11] component in our text mining workflow that accepts mapping rules as input and transforms the designated document's annotations into their equivalent RDF triples. For each annotation type that is to be exported, the mapping rules have an entry that describes: *(i)* the annotation type in the document and its corresponding semantic type, *(ii)* the annotation's features and their corresponding semantic type, and *(iii)* the relations between exported triples and the type of their relation. Given the mapping rules, the mapper component then iterates over the document's entities and exports each designated annotation as the subject of a triple, with a custom predicate and its attributes, such as its features, as the object. Table 2 shows some example mapping rules.

Table 2. Example mapping rules for transforming annotations to RDF triples

Mapping rule examples for Subjects		
Resource	*Type*	*Corresponding class in LOV*
Contribution	annotation type	http://salt.semanticauthoring.org/ontologies/sro#Contribution
Author	annotation type	http://xmlns.com/foaf/0.1/Person

Mapping rule examples for Properties		
Domain	*Range*	*Corresponding property in LOV*
Document	Contribution	http://lod.semanticsoftware.info/pubo/pubo#hasAnnotation
Author	Affiliation	http://purl.org/vocab/relationship/employedBy

3 Implementation

We implemented our text mining pipeline described in Sect. 2 based on the *General Architecture for Text Engineering* (GATE) framework [4]. The pipeline accepts scientific literature in PDF, HTML or plain text format from local or remote URLs as input and stores the extracted entities in form of an RDF document in a knowledge base as output.

3.1 Text Pre-processing

When the input document is in PDF format, we first use Xpdf[12] to extract its textual content into a plain text file. We have observed that the extraction process often introduces erroneous characters to the output text, especially for accented letters. Therefore, in order to prevent cascading such defects to the downstream processing resources, we first normalize the text by replacing faulty character encodings with their correct Unicode. Next, we use GATE's ANNIE plugin [5] to pre-process the document's text into smaller meaningful units, such as word tokens and sentences. The Gazetteer processing resource then generates so-called *Lookup* annotations from word tokens that match entries in its

[11] Originally called the "*RDF Mapper*", it is now an independent open source project available at http://www.semanticsoftware.info/lodexporter.

[12] Xpdf, http://www.foolabs.com/xpdf/.

dictionary. We also use GATE's Morphological Analyzer resource to detect the root form of all word tokens, such as plurals and various verb tenses, so they can be directly matched against the gazetteers terms. Finally, the annotated text is passed onto the Rule Transducer component to classify the document's sentences.

3.2 Rule-Based Extraction of Contextual Entities

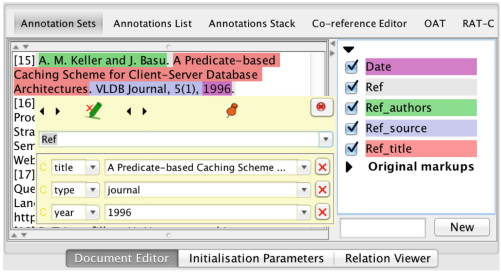
The rules of our pipeline's transducers are implemented using GATE's JAPE language that provides for defining regular expressions over a document's annotations (by internally transforming them into finite-state transducers). The transducing process is conducted in an incremental manner: First, various segments of the document (e.g., Abstract, Main Body, References) are detected so that further analysis can be properly focused, for example, Authors and Affiliations are only detected in the Metadata_body segment of the document. Then, several other JAPE rules are executed sequentially to find Authors, Affiliations, References, Ontology and Funding Agency mentions in text,[13] as described in Sect. 2.2.

For rhetorical entities, multiple JAPE rules are executed sequentially to detect deictic phrases and metadiscourse elements. Finally, depending on the type of the sentence's main verb phrase, the transducer annotates the boundary of the sentence under study with RhetoricalEntity as its type and a reference to the LOV, such as the Contribution class in the SALT Rhetorical Ontology, as its semantic class. Figure 2 shows a sequence of JAPE rules to detect the authors and title of a Reference entity (left) and its corresponding annotation in GATE Developer environment (right).

```
Rule: reference_authors(
 {Person}
 ({Token.kind=="punctuation",Token.string==","}{Person})*
 (({Token.kind=="punctuation",Token.string==","})?
 {Token.string=="and"} {Person})?
):mention
-->
:mention.Ref_authors  = {debugRule = "reference_authors"}

Rule: reference_title(
 {Ref_authors}
 ({Token.string==":"} | {Token.string=="."})
  (({Token, !Token.string=="."})+)?:title
  {Token.string=="."}
):mention
-->
:title.Ref_title  = {content = :title@cleanString}
```

(a) Example JAPE rules



(b) Detected annotations in GATE Developer

Fig. 2. Rule-based extraction of References with JAPE

3.3 Knowledge Base Population

The LODeXporter component is implemented as a GATE processing resource that uses the Apache Jena[14] library to export the document annotations to

[13] Several of our named entity extraction rules are extensions of GATE's ANNIE plugin [5].

[14] Apache Jena, http://jena.apache.org.

RDF triples, according to custom mapping rules, described in Sect. 2.3. The mapping rules themselves are stored in the knowledge base, expressed using RDF triples that explicitly define what annotation types need to be exported and what vocabularies and relations must be used to create a new triple in the knowledge base. Figure 3 shows an excerpt of the mapping rules to export Author and Affiliation annotations and their relations into semantic triples.

```
@prefix map: <http://semanticsoftware.info/mapping#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix cnt: <http://www.w3.org/2011/content#> .
@prefix rel: <http://purl.org/vocab/relationship/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix gn: <http://www.geonames.org/ontology#> .

### Annotation Mapping ###
map:GATEAuthor a map:Mapping ;
       map:type   foaf:Person ;
       map:GATEtype  "Author" ;
       map:hasMapping map:GATEContentMapping .

map:GATEAffiliation a map:Mapping ;
       map:type   foaf:Organization ;
       map:GATEtype  "Affiliation" ;
       map:hasMapping map:GATEContentMapping ;
       map:hasMapping map:GATELocatedInFeatureMapping .

### Feature Mapping ###
map:GATEContentMapping a map:Mapping ;
       map:type   cnt:chars ;
       GATEattribute "content" .

map:GATELocatedInFeatureMapping a map:Mapping ;
       map:type   gn:LocatedIn ;
       GATEfeature "locatedIn" .

### Relation Mapping ###
map:AuthorAffiliationRelationMapping a map:Mapping ;
       map:type rel:employedBy ;
       map:domain map:GATEAuthor ;
       map:range map:GATEAffiliation ;
       GATEattribute "employedBy" .
```

Fig. 3. Excerpt of the mapping rules for exporting Authors, Affiliations and their relations

The mapping rules shown in Fig. 3 describe exporting GATE annotations into several inter-connected triples: Each Author annotation in the document should be exported with `<foaf:Person>` as its type, and its verbatim content in text using the `<cnt:chars>` predicate. Similarly, Affiliation annotations are exported with their "*locatedIn*" feature describing their geographical position from the GeoNames ontology (`<gn:locatedIn>`). Subsequently, the value of the "*employedBy*" feature of each Author annotation is used to construct a `<rel:employedBy>` relation between an author instance and its corresponding affiliation instance in the knowledge base. We used vocabularies from our PUBO ontology wherever no equivalent entity was available in the LOV. For example, we use the `<pubo:containsNE>` property to build a relation between rhetorical entities and the topics that appear within their boundaries (detected by an NER tool).

Ultimately, the LODeXporter processing resource generates all of the desired RDF triples from the document's annotations, and stores them in a scalable, TDB-based[15] triplestore. In addition to the challenge queries, in [1], we demonstrated a number of complex queries that such a semantically-rich knowledge base can answer.

4 Results and Discussion

We analyzed the complete dataset set, consisting of 183 documents (101 in the training set and 82 additional papers for evaluation), with our text mining pipeline and populated the knowledge base in a TDB-based triplestore. The total number of RDF triples generated from processing the complete training set is 506,694, describing the challenge entities, their relations, rhetorical elements, named entities, as well as the triples from the mapping rules. On average, the processing time of extracting and triplification of the knowledge in the proceedings was between 7 and 52 (Mean: 17.30) seconds per volume (running on a 2.3 GHz Intel Core i7 MacBook Pro with 16 GB memory).

Evaluation on Training Set (Pre-Challenge). Prior to release of the testing dataset, we evaluated the performance of our text mining pipeline against a gold standard corpus that we manually curated. We annotated 20 random papers from the training dataset for all of the entity types described in Sect. 2.2 and compared the *Precision*[16] and *Recall*[17] of our pipeline against human judgment. Figure 4 shows the results of our evaluation and the average F1-measure,[18] using GATE's Corpus Quality Assurance tool. In particular, we observed that the precision and recall of the pipeline suffers whenever *(i)* the organization names are

Corpus statistics | Document statistics

Annotation	Match	Only A	Only B	Overlap	Prec.B/A	Rec.B/A	F1.0-a.
Abstract_body	13	2	1	4	0.8333	0.7895	0.8108
Affiliation	23	8	3	9	0.7857	0.6875	0.7333
Author	66	0	3	2	0.9437	0.9853	0.9640
Metadata_body	18	1	1	1	0.9250	0.9250	0.9250
Ref_authors	200	2	29	23	0.8393	0.9400	0.8868
Ref_source	175	12	14	25	0.8762	0.8844	0.8803
Ref_title	192	19	12	14	0.9128	0.8844	0.8984
References_body	10	0	1	9	0.7250	0.7632	0.7436
Title	20	0	0	0	1.0000	1.0000	1.0000
Macro summary					0.8712	0.8733	0.8714
Micro summary	717	44	64	87	0.8762	0.8968	0.8864

Fig. 4. Qualitative analysis of the pipeline performance vs. our gold standard

[15] Apache TDB, http://jena.apache.org/documentation/tdb/.
[16] Precision is the fraction of extracted annotations that are relevant.
[17] Recall is the fraction of relevant annotations that are extracted.
[18] F-measure is the harmonic mean between Precision and Recall.

in a different language than English, *(ii)* authors used unconventional section headers that negatively impacts text segmentation, and *(iii)* anomalies in bibliographical entries were found in text, e.g., arbitrary abbreviation of journal or venue names and author names.

Evaluation on the Complete Set (Post-Challenge). Once the testing set was released for the challenge, we populated our knowledge base with processing the complete dataset of 183 documents (see Sect. 4). We then evaluated the precision (correctness) and recall (completeness) of our populated KB, by comparing the results of our formulated SPARQL queries, shown in Table 3, against the gold standard provided by the challenge coordinators. Posing 50 queries (5 different queries for each of the challenge's 10 queries) against the populated knowledge base yielded an average F-measure of 0.24 (Precision: 0.3, Recall: 0.25). A closer inspection revealed that while our KB performed relatively well in answering **Q2.1**–**Q2.4**, **Q2.9** and **Q2.10** (average F-measure of 0.43), the overall F-measure suffered from zero recall in **Q2.5**,[19] **Q2.6** and **Q2.7** (and obviously, in **Q2.8** since we did not extract any of its required entities).

Table 3. Challenge queries and their equivalent interpretation in our KB (excluding Q2.8)

Challenge query	Equivalent interpretation in our knowledge base
Q2.1: (Authors &) Affiliations in a paper	For the given paper, return all annotations of type `<foaf:Organization>` (affiliations), if it appears as object of the predicate `<rel:employedBy>` and the subject is of type `<foaf:Person>` (author)
Q2.2: Papers from a country	Return all papers that have an annotation of type `<foaf:Organization>` (affiliation), which has a predicate `<gn:locatedIn>` and the object's string value is the given country
Q2.3: Cited works	For the given paper, return all annotations of type `<swrc:Publication>` (reference)
Q2.4: Recent cited works	For the given paper, return all annotations of type `<swrc:Publication>`, which have a predicate `<fabio:hasPublicationYear>` and the object's numerical value is greater than the given year
Q2.5: Cited Journal Papers	For the given paper, return all annotations of type `<swrc:Publication>`, which have a predicate `<ov:category>` and the object's string value is "*journal*"
Q2.6: Research grants	For the given paper, return all annotations of type `<frapo:Grant>`
Q2.7: Funding agencies	For the given paper, return all annotations of type `<frapo:FundingAgency>`
Q2.8: EU projects	*Not addressed in our system*
Q2.9: Related ontologies	For the given paper, return all annotations of type `<owl:Ontology>`
Q2.10: New ontologies	For the given paper, return all annotations of type `<owl:Ontology>`, which have a predicate `<opmw:hasStatus>` and the object's string value is "*new*"

Prefixes used: foaf: <http://xmlns.com/foaf/0.1/>, rel: <http://purl.org/vocab/relationship>, gn: <http://www.geonames.org/ontology#>, swrc: <http://www.geonames.org/ontology#>, fabio: <http://purl.org/spar/fabio/>, ov: <http://open.vocab.org/terms/>, frapo: <http://purl.org/cerif/frapo/>, owl: <http://www.w3.org/2002/07/owl#>, opmw: <http://www.opmw.org/ontology/>

[19] The zero recall for our Q2.5 was due to an error in the mapping rules, where an entity was mapped to two different classes. Apart from that, the annotations were correctly extracted.

5 Conclusions

With the ever-growing amount of information available, students, scientist, and employees spend an ever-increasing proportion of their time searching for the right information. Semantic enrichment of scholarly literature facilitates the automated discovery of knowledge and the integration of data between otherwise disparate documents. The second edition of the Semantic Publishing Challenge aimed at fostering the development of tools for the automatic generation of such metadata. In this context, we described the details of our rule-based text mining system that can extract various semantic information from computer science workshop proceedings. We also introduced a novel, flexible system to transform the detected entities into semantic triples and populate a knowledge base, interlinked with other resources on the Linked Open Data (LOD) cloud. The resulting semantic knowledge base, thus, holds machine-interpretable scientific knowledge that can be exploited through various services, ranging from queries [1] to semantic wikis [6], custom-tailored to a user's task and information needs. In the future, we aim to iteratively improve our text mining pipeline. Working together with challenge organizers and participants, we also hope to address the aggregation of each group's results: Since no data model was enforced in the challenge rules, the individual, submitted results were based on a diverse set of models and vocabularies. A collaboratively generated knowledge base could serve as a unified, clean open dataset for future research and development in semantic publishing initiatives.

References

1. Sateli, B., Witte, R.: What's in this paper? Combining rhetorical entities with linked open data for semantic literature querying. In: Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 2015), Florence, Italy, ACM (2015)
2. Constantin, A., Peroni, S., Pettifer, S., David, S., Vitali, F.: The Document Components Ontology (DoCO). The Semantic Web Journal (2015) (in press). http://www.semantic-web-journal.net/system/files/swj1016_0.pdf
3. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT - semantically annotated LaTeX for scientific publications. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 518–532. Springer, Heidelberg (2007)
4. 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). University of Sheffield, Department of Computer Science (2011)
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002) (2002)
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. CEUR-WS.org (2014)