What's in this paper? Combining Rhetorical Entities with Linked Open Data for Semantic Literature Querying

Bahar Sateli and René Witte
Semantic Software Lab
Department of Computer Science and Software Engineering
Concordia University, Montréal, QC, Canada
[sateli,witte]@semanticsoftware.info

ABSTRACT

Finding research literature pertaining to a task at hand is one of the essential tasks that scientists face on daily basis. Standard information retrieval techniques allow to quickly obtain a vast number of potentially relevant documents. Unfortunately, the search results then require significant effort for manual inspection, where we would rather select relevant publications based on more fine-grained, semantically rich queries involving a publication's contributions, methods, or application domains.

We argue that a novel combination of three distinct methods can significantly advance this vision: (i) Natural Language Processing (NLP) for *Rhetorical Entity* (RE) detection; (ii) *Named Entity* (NE) recognition based on the Linked Open Data (LOD) cloud; and (iii) automatic generation of RDF triples for both NEs and REs using semantic web ontologies to interconnect them. Combined in a single workflow, these techniques allow us to automatically construct a knowledge base that facilitates numerous advanced use cases for managing scientific documents.

1. INTRODUCTION

Modern search engines typically return thousands of scientific articles for a query in a matter of seconds, leaving researchers with the task of manually combing through the results in order to find the information they need – a time-consuming and laborious activity, during which critical knowledge can be easily missed.

To support users in their concrete tasks involving scientific literature, we need to go beyond standard information retrieval methods, such as keyword search. Our vision is to offer support for semantically rich queries that users can ask from a knowledge base of scientific literature, including specific questions about the *contributions* of a publication or the application of specific *methods* for, e.g., an experiment. For example, a user might want to ask the question *"Show me all full papers from the SePublica workshops, which contain a contribution involving 'linked data'."*

We argue that this can be achieved with a novel combination of three approaches: Natural Language Processing (NLP), Linked Open Data (LOD)-based entity detection and semantic vocabularies. By applying NLP techniques for rhetorical entity (RE) recognition to scientific documents, we can detect which text fragments form, e.g., a *contribution*, an *experiment*, or a *claim*. By themselves, they provide for use cases such as summarization, but cannot answer what precisely a contribution is *about*. Manually curating and updating all possible research topics, methods, etc. for NLP detection is not a scalable solution either. However, the *Linked Open Data* (LOD) cloud [6] already provides a continually updated source of a wealth of knowledge across nearly every domain, with explicit and machine-readable semantics. After linking entities detected in research papers 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 data," even when it only mentions "linked open data," since they are semantically related in the DBpedia ontology. However, linked NEs alone again do not help in precisely identifying literature for a specific task: Did the paper actually make a new contribution about "linked data," or just mention it as an application example? Our idea is that by combining the REs with the LOD NEs, we can answer questions like these in a more precise fashion than either technique alone. This requires transforming the NLP results into RDF[1] format, based on a shared vocabulary, so that they can take part in semantically rich queries and ontology-based reasoning.

We performed preliminary experiments, where we demonstrate the feasibility of these ideas by automatically constructing a knowledge base from several years of the *SePublica*[2] workshop proceedings on Semantic Publishing. Note that all queries and results shown in this paper can be verified by visiting the online version of the queries at http://www.semanticsoftware.info/save-sd-2015.

2. FOUNDATIONS

Our work is based on three foundations: NLP techniques for rhetorical entity recognition, named entity recognition in linked open data, and vocabularies for semantic markup of scientific documents.

2.1 Rhetorical Entities

In the context of scientific literature, rhetorical entities (REs) are spans of text (sentences, passages, sections, etc.) in a document, where authors convey their findings, like Claims or Arguments, to the readers. REs are usually situated in certain parts of a document, depending on their role. For example, the authors' Claims are mentioned in the Abstract, Introduction or Conclusion section of a paper, and seldom in the Background. This conforms with the researchers' habit in both reading and writing scientific articles. Indeed, according to a recent survey [13], researchers stated that they are interested in specific parts of an article when searching for literature, depending on their task at hand. Verbatim extraction of REs from text helps to efficiently allocate the attention of humans when reading a paper, as well as improving retrieval mechanisms by finding documents based on their REs (e.g., "*Give me all papers with implementation details*").

Existing works in automatic RE extraction date back to the late 1980s and are mostly based on Mann's *Rhetorical Structure Theory* (RST) [20]. Marcu [11] developed a rhetorical parser that derives the discourse structure from unrestricted text and uses a decision tree to

[1] Resource Description Framework (RDF), http://www.w3.org/standards/techs/rdf

[2] SePublica Workshop, http://sepublica.mywikipaper.org/

extract *Elementary Discourse Units* (EDUs) from text. Teufel [17] identifies so-called *Argumentative Zones* (AZ) from text as a group of sentences with the same rhetorical role. She uses statistical machine learning models and sentential features to extract AZs from a document. Applications of AZs include document management and automatic summarization tasks [17].

In recent years, work on RE recognition has been largely limited to biomedical and chemical documents. The *HypothesisFinder* [10] uses machine learning techniques to classify sentences in scientific literature in order to find speculative sentences. Combined with an ontology to find named entities in text, HypothesisFinder can establish hypothetical links between statements and their concepts in the given ontology. The JISC-funded ART project aimed at creating an "*intelligent digital library*," where the explicit semantics of scientific papers is extracted and stored using an ontology-based annotation tool. The project produced SAPIENT[3] *(Semantic Annotation of Papers: Interface & ENrichment Tool)*, a web-based tool to help users annotate experiments in scientific papers with a set of *General Specific Concepts* (GSC) [8]. The development of SAPIENT was eventually succeeded by the SAPIENTA *(SAPIENT Automation)* tool [7] that uses machine learning techniques to automatically annotate chemistry papers using the ART corpus as the training model.

2.2 Named Entity Linking

An active research area in the Semantic Web community is concerned with recognizing entities in text and linking them with the LOD cloud [6]. This task is related to, but different from named entity (NE) recognition as traditionally performed in NLP in two aspects: First, only entities described on the LOD are discovered (e.g., a city name not present on an LOD source would not be detected, even if an NLP method could identify it as such) and second, each entity must be linked to a unique URI on the LOD cloud (i.e., requiring not just identifying the string "Paris" as a city, but linking it to the correct URI on the web, also known as grounding).

A well-known tool for linked NE detection is DBpedia Spotlight [12, 2], which automatically annotates text with DBpedia resource URIs. It compares surface forms of word tokens in a text to their mentions in the DBpedia ontology. After disambiguating the sense of a token, it creates a link to its corresponding concept in DBpedia. AIDA [21] is an online tool that extracts and disambiguates NEs in a given text by calculating the prominence (frequency) and similarity of a mention to its related resources on the DBpedia, Freebase[4] and YAGO[5] ontologies. More recently, [19] introduced AGDISTIS, a graph-based method that is independent of the underlying LOD source and can be applied to different languages. In their evaluation, it outperformed other existing tools on several datasets.

2.3 Linked Data Vocabularies

In recent years, the Semantic Publishing community increasingly focused on developing vocabularies based on W3C standards, such as RDFS and OWL ontologies, for the semantic description of research publications.

SALT [4] is a framework for the semantic annotation of scientific literature. The SALT framework employs a user-driven approach, where authors manually mark up chunks of text using LaTeX macros with semantic annotations while they are writing a manuscript. It was later extended and adapted for extracting Claims from text with the ultimate goal of creating a knowledge network from scientific publications. Groza et al. introduced ClaiSE [4] and its successor, the KonneX$^{\text{SALT}}$ [5] system, which provide support for (manual) identification, referencing and querying of claims in a collection of documents.

Peroni introduced the EARMARK [3] markup meta-language that models documents as collections of addressable text fragments and associates their content with OWL assertions to describe their structural and semantic properties. He is also the principal author of the DoCO[6] ontology, which is part of the SPAR (Semantic Publishing and Referencing) ontology family [15]. The DoCO ontology specifically defines components of bibliographic documents, like the main matter of books and theses, chapters, figures, and bibliography sections, enabling their description in RDF format.

CoreSC [9] takes on a different approach of annotating scientific documents. It treats scientific literature as a human readable representation of scientific investigations and therefore, has a vocabulary that pertains to the structure of an investigation, like Experiment or Observation. CoreSC is itself a subpart of the EXPO [16] ontology, a comprehensive vocabulary for defining scientific experiments, like Proposition or Substrate. While ontologies like SALT or AZ-II [18] focus on the rhetorical structure of a document, ontologies like CoreSC and EXPO are used for supporting reproducibility in domains, like chemistry or the *omics* sciences.

2.4 Discussion

In our work, we follow an approach similar to Teufel's in that we use NLP techniques for recognizing REs in scientific documents. However, rather than looking at documents in isolation, we aim at creating a linked data knowledge base from the documents, described with common Semantic Web vocabularies and interlinked with other LOD sources, such as DBpedia. We are not aware of existing work that combines NLP methods for RE detection with Semantic Web vocabularies in a fully-automated manner.

Entity linking is a highly active research area in the Semantic Web community. However, it is generally applied on general, open domain content, such as news articles or blog posts, and none of the existing datasets used for evaluation contained scientific publications. To the best of our knowledge, our work is among the first to investigate the application of entity linking on scientific documents, as well as combining LOD entities with rhetorical entities.

3. DESIGN

In this section, we provide a step-by-step description of our approach towards semantic representation of scientific literature (Figure 1). In our approach, users query a knowledge base that is automatically constructed from NLP analysis results (Section 3.1) and interlinked with resource on the LOD cloud (Section 3.2).

3.1 Automatic Detection of REs

We designed a lightweight NLP pipeline to automatically detect rhetorical entities in scientific literature, currently limited to Claims and Contributions. It can classify sentences of a document into one of three categories (Claim, Contribution, or neither) using a rule-based approach.

Our text mining pipeline has multiple gazetteer (dictionary) lists that contain so-called *trigger* words, used to mark corresponding tokens in text for further processing. For example, we have curated a list of general terms used in computer science (30 entries), such as "*framework*" and "*approach*," as well as a comprehensive list of

[3] SAPIENT, http://www.aber.ac.uk/en/cs/research/cb/projects/art/software/

[4] Freebase, https://www.freebase.com

[5] Yet Another Great Ontology (YAGO), http://www.mpi-inf.mpg.de/yago-naga/yago

[6] The Document Components Ontology (DoCO), http://purl.org/spar/doco

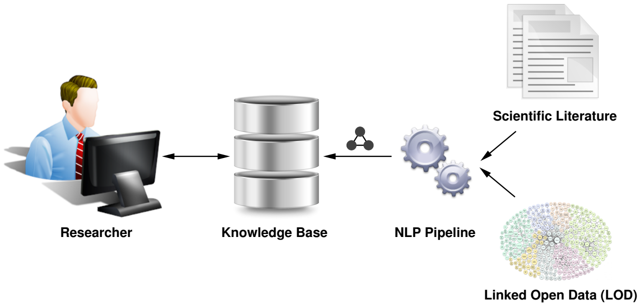


Figure 1: Semantic analysis of scientific literature

verbs used in the scientific argumentation context (160 entries), like "*propose*" and "*develop*," categorized by their rhetorical functions in text. During processing of each document, the pipeline compares each token's root in text against its dictionary of trigger words and produces annotations. We then use rules over the detected annotations, such as text tokens, their part-of-speech and trigger words from the gazetteer lists, to find REs in text. Detection of a rhetorical entity is performed in two incremental steps: First, we detect *metadiscourse* elements in text, i.e., sentences where the authors describe what is being presented in the paper. Metadiscourse entities often contain a discourse *deixis*, such as "*in this paper*" or "*here*," as well as a verb from our gazetteer list of rhetorical verbs. Using hand-crafted rules on sequences of metadiscourse elements and the rhetorical functions of the verbs mentioned in the sentence, we classify each sentence into one of our three categories, like the example shown below:

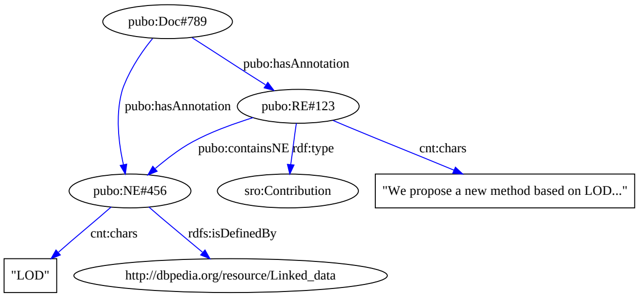
$$\text{Matched string: } \underbrace{\overbrace{\text{In this paper we propose}}^{\texttt{Metadiscourse}} \text{ a method}\ldots}_{\texttt{RhetoricalEntity (Contribution)}}$$

We have designed 11 rules so far to capture various patterns of REs in text, both for Claim and Contribution annotation types, distinguishing Claims as rhetorical entities with a comparative voice or declaration of novelty.

3.2 Automatic Detection of NEs

Using the NLP pipeline described above, we can now find and differentiate REs in a scientific document. However, using REs alone a system is still not able to understand the *topics* being discussed in a document; for example, to generate a topic-focused summary. Therefore, the next step towards constructing a knowledge base of scientific literature is inspecting the named entities that appear in a document. Our hypothesis here is that the extraction of named entities provides the means to represent the main topics being discussed in a paper. Therefore, the detection of the presence of such entities, along with linguistic constituents of the RE fragments, will help towards understanding the meaning of an article's content and position of its authors regarding the detected entities, e.g., 'enhancing algorithm *A*' or 'applying method *M*.'

Instead of applying domain-specific NLP pipelines for NE detection, we want to reuse the LOD cloud as a structured, continually updated source of structured knowledge. To evaluate whether LOD can sufficiently cover NEs in a scientific context, we manually annotated Abstract, Introduction and Conclusion sections of our own paper from *SePublica2014* [14] with NEs as the gold standard. Then, we selected the DBpedia knowledge base[7] and manually inspected whether the entities in the gold standard were present: All of the 55 manually annotated entities in the paper had corresponding resources in the DBpedia knowledge base.



Prefix	Vocabulary	URI
pubo	Our Publication Model	<http://lod.semanticsoftware.info/pubo#>
sro	SALT Rhetorical Onto.	<http://salt.semanticauthoring.org/ontologies/sro#>
rdf	W3C RDF	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs	W3C RDF Schema	<http://www.w3.org/2000/01/rdf-schema#>
cnt	W3C Content Onto.	<http://www.w3.org/2011/content#>
dbpedia	DBpedia Onto.	<http://dbpedia.org/resource/>

Figure 2: Example RDF triples using our publication schema

Based on this experiment, it seems feasible to describe NEs in a scientific publishing context using LOD URIs. To further test this hypothesis, we selected the DBpedia Spotlight annotation tool described in Section 2.2 to automate entity recognition.

3.3 Semantic Representation of Entities

In order to transform the detected rhetorical and named entities into an interoperable and machine-understandable data type that can be added to a semantic knowledge base, we chose to represent all detected entities, as well as some metadata about each document, based on the RDF standard.

We developed a vocabulary for scientific literature constructs that describes a document's various segments (e.g., sentences) and their contained entities, partly by using existing shared vocabularies. We model REs as a subset of document sentences with a specific type, which may in turn contain a list of topics, i.e., named entities with URIs linked to their LOD resources. We chose to reuse the DoCO vocabulary for our experiments, since it has a vocabulary for both structural and rhetorical entities of a document through importing the SALT Rhetorical Ontology. Therefore, using the same ontology, we can describe both the structure of documents (e.g., Abstract, Title), as well as various REs types (e.g., Contributions). We have added our own vocabulary to describe the relations between a document and its contained entities. Our PUBlication Ontology uses "pubo" as its namespace throughout this paper. Figure 2 shows example RDF triples using our publication model and other semantic web shared vocabularies.

For representing the provenance of NLP results and connecting them with the source documents, we also provide for leveraging the Open Annotation (OA)[8] model. However, for the sake of brevity, we only work with a simplified set of generated triples in this paper.

4. IMPLEMENTATION

We now provide some implementation details of our approach described in the previous section.

4.1 Extraction of REs using NLP

Our NLP pipeline is implemented based on the *General Architecture for Text Engineering* (GATE) [1]. We reused GATE's ANNIE pre-processing resources to transform a paper's full text into sen-

[7] DBpedia, http://dbpedia.org

[8] Open Annotation Model, http://www.w3.org/ns/oa

tences and extract the part-of-speech and root form of all text tokens. After pre-processing, the pipeline's gazetteer lists are compared against tokens of the text to produce `Lookup` annotations. Next, finite-state transducers apply the rules described in Section 3.1 to sequences of `Tokens` and their `Lookup` annotations, in order to detect the rhetorical entities. The rules are implemented using GATE's JAPE [1] language that compiles regular expressions into finite-state transducers.

In the transducing phase, we first extract `Metadiscourse` annotations in a text, by detecting a discourse deixis followed by the authors attributions, e.g., the pronoun "*we*." A JAPE rule to extract a Contribution sentence containing a metadiscourse is shown below:

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

The above rule reads: A `Deictic` annotation immediately followed by a pronoun (`PRP`), optionally followed by an adverb (`RB`), immediately followed by an action Lookup annotation, when seen in the text, make up for a `Metadiscourse` annotation. Next, depending on rhetorical type of the sentence's main verb phrase, subsequent rules in the transducer classify the type of the detected RE with one of our custom vocabulary classes and produce `RhetoricalEntity` annotations. In this case, we also defined the detected annotation to be of type Contribution described in the SALT Rhetorical Ontology.

4.2 NE Grounding using DBpedia Spotlight

For NE grounding, we locally installed the DBpedia Spotlight[9] tool [2] version 0.7[10] and used its RESTful annotation service to find and disambiguate NEs in our documents. To integrate the NE detection process in our semantic analysis workflow, we implemented a GATE processing resource (PR) that acts as a wrapper for the Spotlight tool. The processing resources sends the full text of the document to Spotlight with an HTTP request and receives an array of JSON objects as the result. Then, it parses each JSON object into the GATE annotation format and adds a `DBpediaNE` annotation, with a DBpedia URI as its feature, to the document.

To further filter the resulting entities, we align them with noun phrases (NPs), as detected by MuNPEx.[11] This way, phrases like "*service-oriented architecture*" will be extracted as one entity and adverbs or adjectives like "*here*" or "*successful*" will be filtered out.

4.3 Semantic Modeling of Detected Entities

We now have REs and NEs detected in the source documents, but they come in a GATE-specific data structure, i.e., GATE Annotations. We developed another GATE processing resource that uses the Apache Jena[12] framework to export them to RDF triples, according to a custom mapping file that translates GATE annotations and their features to the vocabularies described in Section 3.3.

The mapping rules themselves are also expressed using RDF and explicitly define what annotation types need to be extracted and what vocabularies and relations must be used to create a new triple in the knowledge base. Using this file, each annotation is exported as the subject triple, with a custom predicate and its attributes, such as its features, as the object. Here are some example rules from our mapping file:

```
<rdf:Description rdf:about="map:GATERhetoricalEntity">
   <rdf:type rdf:resource="map:Mapping"/>
   <map:GATEtype>RhetoricalEntity</map:GATEtype>
   <map:hasMapping rdf:resource="map:GATEURIFeatureMapping"/>
</rdf:Description>

<rdf:Description rdf:about="map:GATEURIFeatureMapping">
   <map:GATEfeature>URI</map:GATEfeature>
   <map:type rdf:resource="rdf:type"/>
</rdf:Description>
```

The first triple describes a mapping of the `RhetoricalEntity` GATE annotation type into an RDF triple, in which the subject's type (either sro:Claim or sro:Contribution) is read from the annotation's `URI` feature and described using `rdf:type` as the predicate (cf. Figure 2). Our GATE PR queries the mapping file for the export rules, produces corresponding RDF triples from a document's annotations, and ultimately stores them in a scalable, TDB-based[13] triplestore.

5. EVALUATION

We use four corpora for our experiments, each containing the *SePublica* workshop proceedings from 2011–2014 (28 papers in total). They are analyzed using our system described above and stored in a TDB instance. Table 1 shows the quantitative results of the populated knowledge base. The total number of RDF triples generated is 209,601. On average, the processing time of extracting REs, NEs and the triplification of their relations was 3.58, 2.55, 3.24 and 2.94 seconds per document for the SePublica 2011, 2012, 2013 and 2014 proceedings, respectively; with DBpedia Spotlight annotation taking up to 60–70% of the processing time (running on a standard 2013 quad-core desktop PC).

Particularly interesting is the relation between entities appearing throughout a whole paper (column 'DBpediaNEs') vs. the NEs local to a rhetorical zone (column 'Distinct DBpedia NE/RE'): As can be seen in Table 1, these are between one and two orders of a magnitude lower. This is encouraging for our hypothesis that NEs appearing in a RE can help to semantically query relevant papers.

5.1 NLP Pipeline Intrinsic Evaluation

For the intrinsic evaluation of our NLP pipeline, we performed a preliminary assessment of its performance against a gold standard. We manually annotated all of the SePublica workshop proceedings for rhetorical entities and compared the precision and recall of our pipeline against human judgment. The results showed a 0.68 F-measure on the task of RE detection.

We analyzed the performance of the NLP pipeline against our gold standard and observed that recall suffers whenever the authors' argumentation is described in passive voice.

5.2 Accuracy of NE Grounding with Spotlight

To estimate the accuracy of NE linking to the LOD, we randomly chose 20–50 entities per document for each corpus and manually evaluated whether they are connected to their correct sense in the DBpedia knowledge base, by inspecting their URIs through a Web browser. Out of the 120 entities manually inspected, 82 of the entities had their correct semantics in the DBpedia knowledge base. Overall, this results in 68% accuracy, which confirms our hypothesis that LOD knowledge bases are useful for the semantic description of entities in scientific documents knowledge base.

Our error analysis of the detected named entities showed that Spotlight was often unable to resolve entities to their correct resource

[9] DBpedia Spotlight, http://spotlight.dbpedia.org

[10] with a statistical model for English (en_2+2)

[11] Multi-Lingual Noun Phrase Extractor (MuNPEx), http://www.semanticsoftware.info/munpex

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

[13] Apache TDB, http://jena.apache.org/documentation/tdb/

Table 1: Quantitative analysis of the populated knowledge base

	Size		DBpediaNEs		REs		Distinct DBpediaNE/RE	
Corpus ID	**Docs**	**Sents**	**Occurences**	**Distinct URIs**	**Claims**	**Contributions**	**Claims**	**Contributions**
SePublica2011	7	1619	8186	2083	3	24	23	191
SePublica2012	7	1304	6986	1917	8	24	39	157
SePublica2013	7	1655	8584	2126	3	36	12	212
SePublica2014	7	1566	8821	2012	2	39	24	293
Total	**28**	**6144**	**32577**	**4973**	**16**	**123**	**98**	**853**

(sense) in the DBpedia knowledge base. Spotlight was also unable to resolve acronyms to their full names. For example, Spotlight detected the correct sense for the term "*Information Extraction*", while the term "(IE)" appearing right next to it was resolved to "*Internet Explorer*" instead. By design, this is exactly how the Spotlight disambiguation mechanism works: popular terms have higher chances to be connected to their surface forms. We inspected their corresponding articles on Wikipedia and discovered that the Wikipedia article on Internet Explorer is significantly longer than the Information Extraction wiki page and has 20 times more inline links, which shows its prominence in the DBpedia knowledge base, at the time of this writing. Consequently, this shows that tools like Spotlight that have been trained on the general domain or news articles are biased towards topics that are more popular, which is not necessarily the best strategy for scientific publications.

6. APPLICATION

We published the populated knowledge based described in the previous section using the Jena Fuseki[14] server that provides a RESTful endpoint for SPARQL queries. We now show how the extracted knowledge can be exploited to support a user in her task. As a running example, let us imagine a use case: A user wants to write a literature review from a given set of documents about a specific topic. The prefixes used in the queries in this section can be resolved using the table in Figure 2.

Scenario 1. *A user obtained the SePublica proceedings from the web. Before reading each article thoroughly, she would like to obtain a summary of the contributions of all articles, so she can decide which articles are relevant to her task.*

Ordinarily, the user would have to read all of the retrieved documents in order to evaluate their relevance – a cumbersome and time-consuming task. However, using our approach the user can directly query for the rhetorical type that she needs from the system:

```
SELECT ?paper ?content WHERE {
  ?paper pubo:hasAnnotation ?rhetoricalEntity .
  ?rhetoricalEntity rdf:type sro:Contribution .
  ?rhetoricalEntity cnt:chars ?content } ORDER BY ?paper
```

The system will then show the query's results in a suitable format, like the one shown in Figure 2, which dramatically reduces the amount of information that the user is exposed to, compared to a manual triage approach.

Retrieving document sentences by their rhetorical type still returns REs that may concern entities that are irrelevant or less interesting for our user in her literature review task. Ideally, the system should return only those REs that mention user-specified topics. Since we model both the REs and NEs that appear within their boundaries, the system can allow the user to further stipulate her request. Consider the following scenario:

Scenario 2. *From the set of downloaded articles, the user would like to find only those articles that have a contribution mentioning 'linked data'.*

Table 2: Four example Contributions from papers

Paper ID	Contribution
SePublica2011/ paper-02.xml	*"We describe a feedback loop resulting from the use of nano-publications, give a detailed example, and explain how this can be combined with existing web technologies."*
SePublica2011/ paper-05.xml	*"This position paper discusses how research publication would benefit of an infrastructure for evaluation entities that could be used to support documenting research efforts (e.g., in papers or blogs), analysing these efforts, and building upon them."*
SePublica2012/ paper-03.xml	*"In this paper, we describe our attempts to take a commodity publication environment, and modify it to bring in some of the formality required from academic publishing."*
SePublica2013/ paper-05.xml	*"We address the problem of identifying relations between semantic annotations and their relevance for the connectivity between related manuscripts."*

Table 3: Two example Contributions about 'linked data'

Paper ID	Contribution
SePublica2012/ paper-07.xml	*"We present two real-life use cases in the fields of chemistry and biology and outline a general methodology for transforming research data into Linked Data."*
SePublica2014/ paper-01.xml	*"In this paper we present a vision for having such data available as Linked Open Data (LOD), and we argue that this is only possible and for the mutual benefit in cooperation between researchers and publishers."*

Similar to Scenario 1, the system will answer the user's request by executing the following query against its knowledge base:

```
SELECT DISTINCT ?paper ?content WHERE {
  ?paper pubo:hasAnnotation ?rhetoricalEntity .
  ?rhetoricalEntity rdf:type sro:Contribution .
  ?rhetoricalEntity pubo:containsNE ?ne.
  ?ne rdfs:isDefinedBy dbpedia:Linked_data .
  ?rhetoricalEntity cnt:chars ?content } ORDER BY ?paper
```

The results returned by the system, partially shown in Table 3, are especially interesting. The query not only retrieved parts of articles that the user would be interested in reading, but it also inferred that both "*Linked Open Data*" and "*Linked Data*" named entities have the same semantics, since the DBpedia knowledge base declares an `owl:sameAs` relationship between the aforementioned entities: A full-text search on the papers, on the other hand, would not have found such a semantic relation between the entities.

So far, we showed how we can make use of the LOD-linked entities to retrieve articles of interest for a user. However, the query returned only those articles with REs that contain an NE with a URI exactly matching that of dbpedia:Linked_data. However, by virtue of traversing the LOD cloud using an NE's URI, we can expand the query to ask for contributions that involve dbpedia:Linked_data or any of its *related* subjects. In our experiment, we interpret relatedness as being under the same category in the DBpedia knowledge base. Consider the scenario below:

Scenario 3. *The user would like to find only those articles that have a contribution mentioning topics related to 'linked data'.*

The system can respond to the user's request in three steps: *(i)* First, through a federated query to the DBpedia knowledge base, we find the *category* that dbpedia:Linked_data has been assigned to – in this

[14] Jena Fuseki Server, http://jena.apache.org/documentation/serving_data/

Table 4: Four example Contributions related to 'linked data'

Paper ID	Contribution
SePublica2012/ paper-01.xml	*"In this paper, we propose a model to specify workflow-centric research objects, and show how the model can be grounded using semantic technologies and existing vocabularies, in particular the Object Reuse and Exchange (ORE) model and the Annotation Ontology (AO)."*
SePublica2014/ paper-01.xml	*"In this paper we present a vision for having such data available as Linked Open Data (LOD), and we argue that this is only possible and for the mutual benefit in cooperation between researchers and publishers."*
SePublica2014/ paper-05.xml	*"In this paper we present two ontologies, i.e., BiRO and C4O, that allow users to describe bibliographic references in an accurate way, and we introduce REnhancer, a proof-of-concept implementation of a converter that takes as input a raw-text list of references and produces an RDF dataset according to the BiRO and C4O ontologies.."*
SePublica2014/ paper-07.xml	*"We propose to use the CiTO ontology for describing the rhetoric of the citations (in this way we can establish a network with other works)."*

case, the DBpedia knowledge base returns "*Semantic web*", "*Data management*", and "*World wide web*" as the categories; *(ii)* Then, we retrieve all other subjects which are under the same identified categories; *(iii)* Finally, for each related entity, we look for REs in the knowledge base that mention the related entities within their boundaries. The semantically expanded query is shown below:

```
SELECT ?paper ?content WHERE {
 SERVICE <http://dbpedia.org/sparql> {
      dbpedia:Linked_data <http://purl.org/dc/terms/subject> ?category .
      ?subject <http://purl.org/dc/terms/subject> ?category . }
 ?paper pubo:hasAnnotation ?rhetoricalEntity .
 ?rhetoricalEntity rdf:type sro:Contribution .
 ?rhetoricalEntity pubo:containsNE ?ne.
 ?ne rdfs:isDefinedBy ?subject .
 ?rhetoricalEntity cnt:chars ?content } ORDER BY ?paper
```

The system will return the results, shown in Table 4, to the user. This way, the user receives more results from the knowledge base that cover a wider range of topics semantically related to linked data, without having to explicitly define their semantic relatedness to the system. This simple example is a demonstration of how we can exploit the wealth of knowledge available in the LOD cloud. Of course, numerous other queries now become possible on scientific paper, by exploiting other linked LOD sources.

7. CONCLUSION

We all need better ways to manage the overwhelming amount of scientific literature available to us. Our approach is to create a semantic knowledge base that can supplement existing repositories, allowing users fine-grained access to documents based on querying LOD entities and their occurrence in rhetorical zones. We argue that by combining the concepts of REs and NEs, enhanced retrieval of document becomes possible, for example, finding all contributions on a specific topic or comparing the similarity of papers based on their REs. To demonstrate the feasibility of these ideas, we developed an NLP pipeline to fully automate the transformation of scientific documents from free-form content, read in isolation, into a queryable, semantic knowledge base.

In future work, we plan to further improve both the NLP analysis and the LOD linking part of our approach. As our experiments showed, general-domain NE linking tools, like DBpedia Spotlight, are biased toward popular terms, rather than scientific entities. Here, we plan to investigate how we can adapt existing or develop new entity linking methods specifically for scientific literature. Finally, to support end users not familiar with semantic query languages, we plan to explore user interfaces and interaction patterns, e.g., based on our *Zeeva* semantic wiki [14] system.

8. REFERENCES

[1] H. Cunningham et al. *Text Processing with GATE (Version 6)*. University of Sheffield, Department of Computer Science, 2011.

[2] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proc. of the 9th Intl. Conf. on Semantic Systems (I-Semantics)*, 2013.

[3] A. Di Iorio, S. Peroni, and F. Vitali. Towards markup support for full GODDAGs and beyond: the EARMARK approach. In *Proceedings of Balisage: The Markup Conference*, 2009.

[4] T. Groza, S. Handschuh, K. Möller, and S. Decker. SALT – Semantically Annotated LaTeX for Scientific Publications. In *The Semantic Web: Research and Applications*, LNCS, pages 518–532. Springer, 2007.

[5] T. Groza, S. Handschuh, K. Möller, and S. Decker. KonneX^{SALT}: First Steps Towards a Semantic Claim Federation Infrastructure. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web: Research and Applications*, volume 5021 of *LNCS*, pages 80–94. Springer Berlin Heidelberg, 2008.

[6] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis lectures on the semantic web: theory and technology. Morgan & Claypool Publishers, 2011.

[7] M. Liakata, S. Saha, S. Dobnik, C. R. Batchelor, and D. Rebholz-Schuhmann. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000, 2012.

[8] M. Liakata and L. Soldatova. Guidelines for the annotation of general scientific concepts. Technical report, Aberystwyth University, 2008. JISC Project Report, http://ie-repository.jisc.ac.uk/88.

[9] M. Liakata, S. Teufel, A. Siddharthan, and C. R. Batchelor. Corpora for the Conceptualisation and Zoning of Scientific Papers. In *LREC*, 2010.

[10] A. Malhotra, E. Younesi, H. Gurulingappa, and M. Hofmann-Apitius. 'HypothesisFinder:' A Strategy for the Detection of Speculative Statements in Scientific Text. *PLoS computational biology*, 9(7):e1003117, 2013.

[11] D. Marcu. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 365–372. Association for Computational Linguistics, 1999.

[12] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proc. of the 7th International Conf. on Semantic Systems*, pages 1–8. ACM, 2011.

[13] A. Naak, H. Hage, and E. Aimeur. Papyres: A Research Paper Management System. In *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on*, pages 201–208, July 2008.

[14] B. Sateli and R. Witte. Supporting Researchers with a Semantic Literature Management Wiki. In *The 4th Workshop on Semantic Publishing (SePublica 2014)*, volume 1155 of *CEUR Workshop Proceedings*, Anissaras, Crete, Greece, May 25 2014.

[15] D. Shotton, K. Portwin, G. Klyne, and A. Miles. Adventures in semantic publishing: exemplar semantic enhancements of a research article. *PLoS Computational Biology*, 5(4):e1000361, 2009.

[16] L. N. Soldatova, A. Clare, A. Sparkes, and R. D. King. An ontology for a Robot Scientist. *Bioinformatics*, 22(14):e464–e471, 2006.

[17] S. Teufel. *The Structure of Scientific Articles: Applications to Citation Indexing and Summarization*. Center for the Study of Language and Information, 2010.

[18] S. Teufel, A. Siddharthan, and C. R. Batchelor. Towards Discipline-independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics. In *EMNLP*, pages 1493–1502, Stroudsburg, PA, USA, 2009. ACL.

[19] R. Usbeck, A.-C. Ngonga Ngomo, S. Auer, D. Gerber, and A. Both. AGDISTIS - Graph-Based Disambiguation of Named Entities using Linked Data. In *International Semantic Web Conference (ISWC)*, LNCS. Springer, 2014.

[20] M. William and S. Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[21] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. AIDA: An online tool for accurate disambiguation of named entities in text and tables. *Proc. VLDB*, 4(12):1450–1453, 2011.