Semantic Annotation of Data Processing Pipelines in Scientific Publications

Sepideh Mesbah(✉), Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben

Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
{s.mesbah,k.fragkeskos,c.lofi,a.bozzon,g.j.p.m.houben}@tudelft.nl

Abstract. Data processing pipelines are a core object of interest for data scientist and practitioners operating in a variety of data-related application domains. To effectively capitalise on the experience gained in the creation and adoption of such pipelines, the need arises for mechanisms able to capture knowledge about datasets of interest, data processing methods designed to achieve a given goal, and the performance achieved when applying such methods to the considered datasets. However, due to its distributed and often unstructured nature, this knowledge is not easily accessible. In this paper, we use (scientific) publications as source of knowledge about Data Processing Pipelines. We describe a method designed to classify sentences according to the nature of the contained information (i.e. scientific objective, dataset, method, software, result), and to extract relevant named entities. The extracted information is then semantically annotated and published as linked data in open knowledge repositories according to the DMS ontology for data processing metadata. To demonstrate the effectiveness and performance of our approach, we present the results of a quantitative and qualitative analysis performed on four different conference series.

1 Introduction

Data is now at the centre of almost all fields of technology and science. Data processing workflows (or "pipelines") facilitate the creation, integration, enrichment, and analysis (at scale) of heterogeneous data, thus often opening the field for before unseen innovation. It comes with little surprise that the scientific community is devoting an increasing amount of attention to the design and testing of data processing pipelines, and to their application and validation to big, and open, data sources.

In scientific publications, scientists and practitioners *share* and *seek* information about the properties and limitations of (1) data sources; and (2) of data processing methods (e.g. algorithms) and their implementations. For instance, a researcher in the field of urban planning could be interested in *discovering state of the art methods for point of interest recommendation (e.g. matrix factorisation) that have been applied to geo-located social media data (e.g. Twitter) with good accuracy results.*

E. Blomqvist et al. (Eds.): ESWC 2017, Part I, LNCS 10249, pp. 321–336, 2017.
DOI: 10.1007/978-3-319-58068-5_20

A system able to answer the query above requires access to a structured representation of the knowledge contained in one or more scientific publication repositories. For instance, it should be possible to *access* and *relate* information about: (1) the objective of a given scientific work; (2) the datasets employed in the work; (3) the methods (i.e. algorithms) and tools (e.g. software) developed or used to process such datasets; and (4) the obtained results.

Our vision is to offer support for semantically rich queries focusing on different aspects of data processing pipelines (e.g. methods, datasets, goals). The availability of a semantically rich, interlinked, and machine readable descriptions (metadata) of such knowledge could provide great benefits in terms of retrieval quality, but also for analysing and understanding trends and developments.

Manually inspecting and annotating papers for metadata creation is a non-trivial and time-consuming activity that clearly does not scale with the increasing amount of published work. Alas, scientific publications are also difficult to process in an automated fashion. They are characterised by structural, linguistic, and semantic features that are different from non-scientific publications (e.g. blogs). In this context, general-purpose text mining and semantic annotation techniques might not be suitable analysis and tools. As a consequence, there is a clear need for methodologies and tools for the extraction and semantic representation of scientific knowledge. Recent work focused on methods devoted to the automatic creation of semantic annotations for text snippets, with respect to either structural [8,10,11], argumentative [12,14], or functional [4,6,7] components of a scientific work. However, to the best of our knowledge, there has been no work yet focusing on extracting metadata focusing on properties of data processing pipelines. Therefore, in this paper, we provide the following contributions:

– A novel approach for the classification of text related to data processing pipelines from scientific publications, and for the extraction of named entities. The approach combines distant supervision learning on rhetorical mentions with named entity recognition and disambiguation.

Our system automatically classifies sentences and named entities into five categories (objectives, datasets, methods, software, results). Sentence classification attains an average accuracy of 0.80 and average F-score 0.59.

– A quantitative and qualitative evaluation of the implementation of our approach, performed on a corpus of 3,926 papers published in 4 different conference series in the domain of Semantic Web (ESWC), Social Media Analytics (ICWSM), Web (WWW), and Databases (VLDB).

We provide evidence of the amount and quality of information on data processing pipelines that could be extracted, and we show examples of information needs that can now be satisfied thanks to the availability of a richer semantic annotation of publications' text.

– The annotations resulting from the evaluation are published in an RDF repository, available for query.[1] We employ the DMS [17] ontology to encode properties related to the objectives, datasets, methods, software, and results described in a scientific publication, and then represent them as RDF graphs.

The remainder of the paper is organised as follows: Sect. 3 introduces the DMS ontology; Sect. 4 describes the data processing pipelines knowledge extraction workflow; Sect. 5 reports the results of the evaluations; Sect. 2 describes related work. Finally, Sect. 6 presents our conclusions.

2 Related Work

In the last few years there has been a growing interest in the open and linked publication of metadata related to scientific publications. There are now several ontologies devoted to the description of scholarly information (e.g. SWRC,[2] BIBO,[3] DMS [17]). The Semantic Dog Food [2] and the RKBExplorer [3] are examples of projects devoted to the publication of "shallow" meta data about conferences, papers, presentations, people, and research areas. A large portion of such shallow metadata is already explicitly given by the authors as part of the final document, such as references, author names, keywords, etc. Still, the extraction of that metadata from a layouted document is complex, requiring specialized methods [19] being able to cope with the large variety of layouts or styles used in scientific publication. In contrast, "deep" metadata as for example the topic, objectives, or results of a research publication pose a greater challenge as such information is encoded in the text itself. The manual creation of such metadata related to scientific publications is a tedious and time-consuming activity. Semi-automatic or automatic metadata extraction techniques are viable solutions that enable the creation of large-scale and up-to-date metadata repositories. Common approaches focus on the extraction of relevant entities from the text of publications by means of ruled-based [11,14], machine learning [8], or hybrid (combination of rule based and machine learning) [6,7] techniques.

These approaches share a common assumption: as the number of publications dramatically increases, approaches that exclusively rely on dictionary-based pattern matching (possibly based on pre-existing knowledge bases) are of limited effectiveness. Rhetorical entities (REs) detection [9] is a class of solutions that aims at allowing the identification of relevant entities in scientific publications by analysing and categorising spans of text (e.g. sentences, sections) that contain information related to a given structural [8,10,11] (e.g. Abstract, Introduction, Contributions, etc.), argumentative [12,14] (e.g. Background, Objective, Conclusion, Related Work and Future Work), or functional (e.g. datasets [4], algorithms [6], software [7]) classification.

[1] Companion website: http://www.wis.ewi.tudelft.nl/eswc2017.
[2] http://ontoware.org/swrc/.
[3] http://bibliontology.com.

In contrast to existing literature, our work focuses on rhetorical mentions that relate to the description (Objective), implementation (Dataset, Method, Software), and evaluation (Result) of data processing pipelines. Thanks to a distant supervision approach and a simple feature model (bags-of-words), our method does not require prior knowledge about relevant entities [4] or grammatical and part-of-speech characteristics of rhetorical entities [6]. In addition, while in previous work [10,11] only one or few sections of the paper (e.g. abstract, introduction) are the target of rhetorical sentences classification, we make no assumption about the location of relevant information. This adds additional classification noise, due to the uncontrolled context of training sentences: it is more likely for a "Result" section to describe experimental results than for a "Related Work" section, where the likelihood of misclassification is higher [9].

3 The DMS Ontology

The `DMS` (Dataset, Method, Software) ontology [17] is designed to support the description and encoding of relevant properties of data processing pipelines, while capitalising on established ontologies. `DMS` has been created in accordance to the *Methondology* guidelines. It has been implemented using OWL 2 DL, and it consists of 10 classes and 30 properties. `DMS` captures five main concepts, namely *objectives*, *datasets*, *methods*, *software*, and *results*.

In the following, we refer to this initial ontology as `DMS-Core`. We provide an overview of the five aforementioned core concepts in Fig. 1 (in order to keep compatibility with existing ontologies, for some concepts, we adopt slightly different naming conventions within the ontology and in this text, i.e., *dataset* is encoded as *disco:DataFile* in `DMS`). Data processing pipelines are composed of one or more methods (*deo:Methods*), and are typically designed and evaluated in the context of a scientific experiment (*dms:Experiment*) described in a publication (*dms:Publication*). An experiment applies data processing methods, implemented by software (*ontosoft:Software* [13]), to one or more datasets (*disco:DataFile*) in order to achieve a given objective (*dms:Objective*), yielding one or more results (*deo:Results*). In each experiment, different implementations or configurations of a method (*dms:MethodImplementation*) or software (*dms:softwareconfiguration*) can be used. However, in this work, we only focus on the core concepts ignoring configurations and implementations.

Our main contribution in this paper is a methodology for the automatic extraction of metadata in accordance with the five core concepts of `DMS`: objective, dataset, method, software, and result. We reach this goal by labeling each of the sentences in a publication when it contains a *rhetorical mention* of one of the five `DMS` concepts. To capture knowledge on the properties and results of this extraction process, we introduce an auxiliary module `DMS-Rhetorical` (Fig. 1) extending `DMS-Core` as discussed in the following. `DMS-rhetorical` allows to link any *dms:CorePipelineConcept* (i.e. the supertype of *objective*, *dataset*, *method*, *software*, and *result*) to an extracted rhetorical mention.

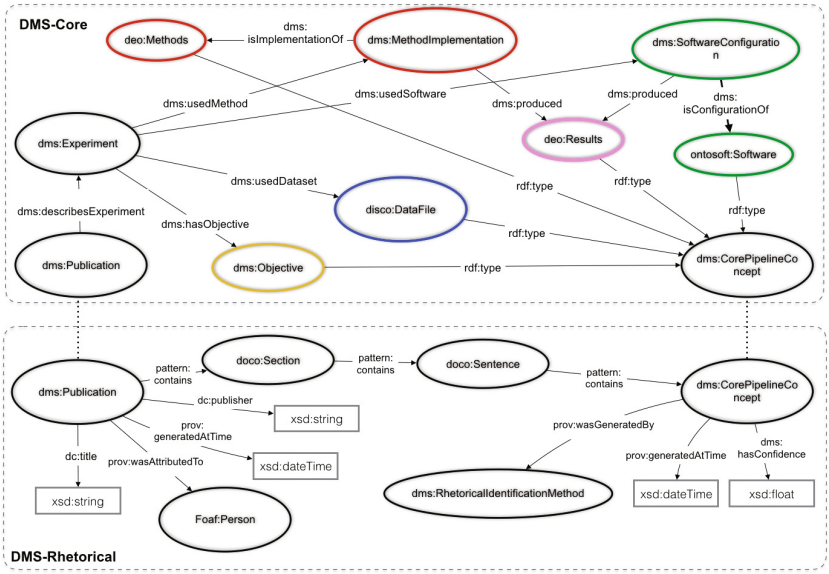

Fig. 1. `DMS-Core` ontology and the `DMS-Rhetorical` extension.

This link includes relevant provenance information such as the source of that mention (e.g. the sentence and section within a publication), but also metadata related to the extraction process, such as the classifier used to associate a sentence to a given `DMS` concept, and the related classification confidence.

We reuse the DoCo [1] ontology for encoding the information on sections and sentences. For each publication, we keep its general metadata including *id*, *title*, *authors*, *year of publication*, and *publisher*. The publication contains (*pattern:contains*) sections and each section of the paper contains several sentences. We store the text of the sentence using the *doco:Sentence* class and link the sentence *pattern:contains* to its *dms:CorePipelineConcept*.

4 DPP Knowledge Extraction Workflow

This section presents the knowledge extraction workflow designed to identify and annotate information referring to data processing pipelines (DPP) along the lines of the main classes of the `DMS` ontology (i.e. datasets, methods, software, results, and objectives). Our whole approach is summarized in Fig. 2. First, we identify rhetorical mentions of a `DMS` main class. In this work, for the sake of simplicity, rhetorical mentions are sought at sentence level. Future works will introduce dynamic boundaries, to capture the exact extent of a mention. Then, we extract named entities from the rhetorical mentions. These entities are filtered and, when applicable, linked to pre-existing knowledge bases, creating the final knowledge repository.

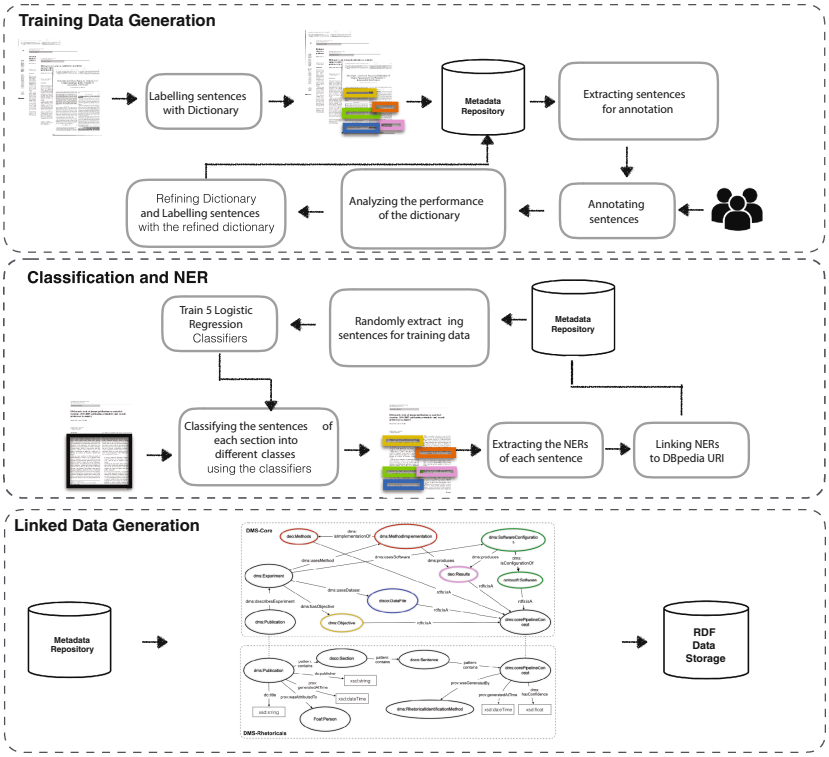

Fig. 2. Data processing pipeline knowledge extraction workflow.

The identification of rhetorical mentions is obtained through a workflow inspired by distant supervision [20], a training methodology for machine learning algorithms that relies on very large, but noisy, training sets. The training sets are generated by means of a simpler classifier, which could rely, for instance, on a mix of expert-provided dictionaries and rules, refined with manual annotations. Intuitively, the training noisiness could be cancelled out by the huge size of the semi-manually generated training data. This method requires significantly less manual effort, while at the same time retaining the performance of supervised classifiers. Furthermore, this approach is more easily adapted to different application domains and changing language norms and conventions.

4.1 Training Data Generation

Data Preparation. Scientific publications, typically available in PDF, are processed using one of the best-state-of-art extraction engines, GeneRation Of BIbliographic Data (`GROBID`) [18,19]. `GROBID` extracts a structured full-text representation as Text Encoding Initiative(TEI)-encoded documents, thus providing easy and reliable access paragraphs and sentences.

Dictionary-Based Sentence Annotation. Our goal is to classify each sentence of a given publication with respect to the five main classes of the DMS Ontology (datasets, methods, software, results, and objectives), based on the presence of rhetorical mentions that are related to such classes. Sentence classification could be obtained by means of a traditional supervised machine learning approach, assuming the presence of a large enough training set of sentence-level annotations. In our previous work [17], we manually created a small set of high-quality sentence-level annotations, relying on expert feedback. However, the annotation of a single publication took around 30–60 min per annotator, showing that this approach was not sufficiently scalable. We therefore opted for a workflow inspired by *distant supervision*. All sentences in our corpus were automatically labeled using a lower-quality and noisy dictionary-based classifier and simple heuristic rules, which are created using the following two-steps approach:

- **Reuse of generic scientific rhetorical phrases**: We relied on manually curated and published dictionaries of phrases and words found in [15,16] as an initial starting point to build our own dictionary. Both papers are writing guides giving advise on how to write an academic text based on best practices and commonly used phrases. [16] covers common phrases for introducing different sections in academic literature, e.g. the abstract, problem statement, methodology, or result discussion. [15] presents an extensive manual corpus study on different parts of scientific argumentation, and gives suggestion for accepted and often used phrases split by different disciplines and publication types.
- **Manual refinement and adaptation to the DMS domain**: The set of dictionary words based on [15,16] did not focus specifically on rhetorical mentions of data processing pipelines (even though classes like "result discussion" are quite related). Therefore, we manually refined those dictionaries and adapted them specifically to our 5 DMS classes. This refinement is based on the careful inspection of 20 papers selected from four Web- and data- related conferences series (ESWC, VLDB, ICWSM, and WWW).

The outcome of these two steps is a more class-specific set of dictionaries. For example the rhetorical phrases "*we collected*" and "*we crawled*" indicate a rhetorical mention of the *dataset* class. We used the dictionary to label sentences of 10 publications randomly selected from the four conferences series, to manually check the performance of the dictionary. For instance, we observed that the word "*data*" alone in a sentence is not a good indicator for being related to *dataset*. However if the word "*data*" co-occurs with "*from*", a relationship with *dataset* is more likely. Several iterations of this manual refinement process lead to the final dictionary used for the following steps. Some example phrases are shown in Table 1.[4] Note that rhetorical mentions used in our refined dictionary are in fact skip n-grams, i.e. we do not expect the terms of each skip n-gram to be adjacent in a sentence (e.g. the rhetorical mention "the aim of this study" stripped of stop words becomes the skip n-gram "aim study").

[4] The dictionaries are available at http://www.wis.ewi.tudelft.nl/eswc2017.

Table 1. Excerpt of dictionary of phrases used for classifying sentences

Objective	*this research, this article, aim study, aim article, purpose paper, we aim, we investigate*
Dataset	*dataset, datasource, data source, collected from, database, collect data, retrieve data*
Method	*we present, we develop, we conduct, we propose, methodologies, method, technique*
Sofware	*tool, obtained using, collected using, extracted using, software*
Result	*we find, shows, show, shown, showed, we found, figure, table, we observe, we compare*

Test and Training Data Generation. We created reliable test and training datasets for both training and benchmarking machine learning classifier as follows. By using the phrases dictionary described in the previous subsection, we label all sentences of all research papers collected with appropriate class labels. Most sentences will not receive a label (as they do not contain any rhetorical mentions), but some may obtain multiple labels. This is for instance common for sentences found in an abstract, which often contain information on *dataset*s, but also on *method*s, or even *result*s. Then, we randomly select a balanced set of sentences with rhetorical mentions of all five classes, and manually inspect the assigned labels. We reclassify them using expert feedback from several annotators, if the pattern-based classifier assigned incorrect labels. Using this approach, we can create a reliable manually annotated and balanced test dataset quicker and cheaper compared to annotating whole publications or random sentences, as the pattern-classifier usually delivers good candidate sentences. Furthermore, this approach allows us to further refine and improve the dictionary by incorporating the expert feedback, allowing us to cheaply re-annotate the whole corpus using the dictionary with higher accuracy compared to the initial classifier.

We assessed the performance of both the dictionary-based classifier and our annotators to decide on the number of manual annotations needed for a reliable test set. We randomly selected 100 sentences from each of the five classes (i.e. 500 in total). Two expert annotators manually checked the assigned labels (a task which was perceived easier by the annotators than applying labels to a random unlabeled sentence). The inter-annotator agreement using the Cohen's kappa measure averaged over all classes was .58 (the Cohen's kappa measures of the individual classes are *objective*: .71, *dataset*: .68, *software*: .37, *result*: .61, and *method*: .53).

4.2 Classification and NER

Machine-Learning-Based Rhetorical Detection. As a second part of our distant supervision workflow, we now train a simple binary Logistic regression classifier for each of the classes using simple `TF-IDF` features for each sentence.

This simple implementation serves as a proof of concept of our overall approach, and can of course be replaced by more sophisticated features and classifiers in future work.

As a test set, we use the 500 sentences (100 per class) manually labeled with their DMS class by our expert annotators. We associated a single label (some sentences can have multiple labels) to each sentence, decided by a simple majority vote. In order to generate the training data for each class, we randomly selected 5000 positive examples from the sentences labeled with that class by the dictionary-based classifier. We also randomly select 5000 negative examples from sentences which are not labeled with that class by the dictionary classifiers. Sentences from the test set were excluded from the pool of candidate training sentences.

Named Entity Extraction, Linking, and Filtering. In the last step of our method, we extract named entities from the sentences that are classified as related to one of the five main DMS classes, filtering out those entities that are most likely not referring to one of the DMS classes, and retaining the others as an extracted entity of the class matching the sentence label.

Named entity extraction has been performed using the TextRazor API[5]. TextRazor returns the detected entities, possibly decorated with links to the DBpedia or Freebase knowledge bases. As we get all named entities of a sentence, the result list contains many entities which are not specifically related to any of the five classes (e.g. entities like "software", "database"). To filter many of these entities, and after a manual inspection, we opted for a simple filtering heuristic. Named entities are assumed to be not relevant if they come from "common" English language (like software, database), while relevant entities are terms referring to domain-specific terms or specific acronyms (like SVM, GROBID, DMS, Twitter data). The heuristic is implemented as look-up function of each term in *Wordnet*.[6] Named entities that can be found in WordNet are removed. As WordNet is focusing on general English language, only domain-specific terms remain. We present the results of the analysis performed on the quality of the remaining named entities in Sect. 5.

4.3 Linked Data Generation

As a final step, we build a knowledge repository based on the DMS-Core and DMS-Rhetorical ontology (outlined in Sect. 3). The repository is populated with classified sentences, and with the lists of entities for each DMS main class, with links to the sentence where each single entity has been detected. Sentences are linked to the containing publications.

[5] http://www.textrazor.com/.
[6] http://wordnet.princeton.edu/.

Listing 1.1 shows an example of a part of an output RDF. The relationships shown in the RDF snippet are from the domain-specific DMS ontology for describing data-processing research. They have not been extracted automatically, as the scope of this work is not on the automatic extraction of relationships between entities.

```
PREFIX doco: <http://purl.org/spar/doco>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX disco: <http://rdf-vocabulary.ddialliance.org/discovery#>
PREFIX dms: <https://github.com/mesbahs/DMS/blob/master/dms.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pattern: <http://www.essepuntato.it/2008/12/pattern>
[a dms:Publication;
dms:describesExperiment  dms:Ncdec5e68ed864a3a24].
dms:Ncdec5e68ed864a3a24   a  dms:Experiment;
      dms:usedDataset [ a   disco:dataFile ;
                        rdf:type      dms:Ncdec5e68ed864a ;
                        prov:value     "Billion Triple Challenge (BTC)"].
dms:Ncdec5e68ed864a   a   dms:CorePipelineConcept;
       pattern:isContainedBy  doco:Ncdec5e68edghgf99.
doco:Ncdec5e68edghgf99 a doco:Sentence;
       prov:value "In our experiments we used real data that were taken from the Billion
             Triple Challenge (BTC) dataset.";
       pattern:isContainedBy doco:Ncdec5ehfdjk67.
doco:Ncdec5ehfdjk67 a doco:Section;
       prov:value "Introduction".
```

Listing 1.1. Example of output RDF: A paper describes an experiment which uses a dataset called (BTC). (BTC) is a CorePipelineConcept linked to sentence of the paper.

5 Evaluation

In this section, we analyse the performance of our metadata extraction pipeline in both a quantitative and qualitative fashion. We focused on four major conference series from different communities with notable scientific contributions to data processing pipelines (Table 2): the European Semantic Web Conference (ESWC), International Conference On Web and Social Media (ICWSM), International Conference on Very Large Databases (VLDB), and the International World Wide Web Conference (WWW). We further present the results of both the dictionary-based and logistic regression-based sentence classifiers on the manually annotated test data. Finally, we analyse and discuss the quality of the entities extracted from the classified sentences.

5.1 Dataset

Table 2 summarises the properties of the experimental dataset, including its size, the number of rhetorical mentions extracted for each class (as decided by the regression-based classifier), and the number of unfiltered unique named entities extracted from the rhetorical mentions taken from scientific publications of a particular conference series. The table shows that methods are the most frequent encountered class, followed by datasets. Table 3 summarises statistics on extracted entities as described in the previous section per class (including

Table 2. Quantitative analysis of the rhetorical sentences and named entities extracted from four conference series. Legend: PAP (papers), SNT (sentences), OBJ (objective), DST (dataset), MET (method), SWT (software), RES (results)

Conf.	Size		Rhetorical sentences					Unique named entities				
	#PAP	#SNT	#OBJ	#DST	#MET	#SWT	#RES	#OBJ	#DST	#MET	#SWT	#RES
ESWC	620	129760	12725	13528	26337	9614	22245	4197	4910	6987	4557	6416
ICWSM	793	52094	6096	4277	8936	1830	13848	2830	2241	3658	1538	4499
VLDB	1492	396457	26953	49855	68336	11919	84662	7301	12052	13920	5741	15959
WWW	1021	253401	23378	19783	49331	10293	58212	6616	6499	10793	5164	11869

Table 3. Number of Named Entities after filtering using the Wordnet.

Conf.	Distinct NER with URI					Distinct NER no URI				
	#OBJ	#DST	#MET	#SWT	#RES	#OBJ	#DST	#MET	#SWT	#RES
ESWC	1157	1206	1779	1200	1454	1874	2427	3497	2193	3219
ICWSM	727	555	944	443	1027	1110	900	1588	519	1974
VLDB	1528	2313	2516	1365	2395	3800	6963	8393	2804	10288
WWW	1990	1630	2904	1613	2860	2742	3153	5382	2148	6247

Table 4. Top-5 most frequent methods applied to IMDB dataset.

ESWC	ICWSM	VLDB	WWW
Semantic Web	LDA	Tuple	Web Page
Sem-CF	Classifier_I	XML	Login
User Modeling	SetLock	Query Plan	Faceted Search
Recommender System	Hashtag	XsKetch	Recommender System
FactBox	Future tense	LS-B	Source Rank

filtering and pruning entities using a Wordnet look-up). Furthermore, we report how many of those entities could be linked to Wikipedia by the TextRazor API (columns *with URI*), thus distinguishing well-known entities (e.g. Facebook, Greedy algorithm) from the newly presented or less popular entities (e.g. SIFT Netnews, RW ModMax. columns *no URI*).

Qualitative Analysis. In this section, we showcase how our approach can be used to fulfill a hypothetical information need of a data scientist, namely: *Which methods are commonly applied to a given data set?*

As an example, we use the popular IMDB dataset of movies and actors, and manually inspect the list of top-6 most frequent methods applied to that dataset in publications grouped by their conference series. The results are shown in Table 4, hinting at the different interests conference venues have for that dataset: ignoring the false positives (like "Web Page" or "XML" - we further discuss false positives later in this section), VLDB as a database-centric conference covers methods like XsKetch (summarisers for improving query plans

in XML databases) or LSB-Trees for better query plans for nearest-neighbour queries, using the IMDB dataset as a large real-life dataset for evaluation database queries; ICWSM with a focus on Social Media research features LDA topic detection and generic classification to analyse IMDB reviews, while ESWC and WWW are interested in recommendations and user modelling.

5.2 Analysis of Rhetorical Classifiers

In the following, we present the results of both the *dictionary-based* and *logistic regression-based* classifiers on the manually annotated test set, summarised in Table 5, relying on commonly used measurements for accuracy, precision, recall, and F-Score. It can be observed that using logistic regression increases the recall for most classes, while having a slightly negative impact on the precision, showing that this approach can indeed generalise from the manually provided dictionaries to a certain extent.

We believe that better performance can be achieved by employing more sophisticated features and classifiers. Furthermore, the performance gains of the logistic regression classifier come for "free" as we only invested time and effort to train the dictionary-based classifier. The best results are achieved for the *Method* class with F-score = 0.71. We manually inspected the sentences labeled as *Software* and *Dataset* to understand reasons for the comparatively low performance of those classes. To certain extend, this can be attributed to the ambiguity of some n-grams in the dictionary. For example, the word *tool* appearing in different sentences can result to misleading labels: e.g., "extraction tool Poka" is about software, but "current end-user tools" is a general sentence not specifically about a software. Similarly confusion can be observed for the word *dataset* for the *Dataset* class. For instance, "twitter dataset" and "using a dataset of about 2.3 million images from Flickr" are labeled correctly, but "quadruple q and a dataset d" is labeled incorrectly. Thus, we conclude that many terms used in *Software* and *Dataset* are too generic (e.g. dataset, tool, database) leading to higher recall, but having a negative impact on precision, demanding more refined rules in our future work.

Table 5. Estimated Accuracy, Precision, Recall and F-score on manually annotated sentences for Dictionary and Logistic regression based classification

	Dictionary based				Logistic regression based			
Classes	Accuracy	Precision	Recall	F-Score	Accuracy	Precision	Recall	F-Score
Objective	0.85	0.49	0.81	0.61	0.84	0.49	0.81	0.61
Dataset	0.84	0.46	0.68	0.55	0.80	0.41	0.81	0.54
Method	0.76	0.79	0.61	0.69	0.76	0.76	0.67	0.71
Software	0.83	0.39	0.52	0.45	0.84	0.34	0.72	0.46
Result	0.84	0.60	0.68	0.63	0.81	0.53	0.71	0.60

5.3 Quality of Extracted Entities

We studied the performance of the Named Entity (NE) extraction modules of our method by means of a mixed quantitative and qualitative analysis. We calculated the Inverse Document Frequency (IDF) of each named entity NE_i extracted from the corpus. IDF is a measure of informativeness, calculated as $IDF(NE_i) = log\frac{|Sentences|}{|NE_i|}$, that is, the logarithmically scaled inverse fraction of the number of sentences in the corpus and the number of sentences containing NE_i. Figure 3 depicts the distribution of IDF values for each NE in the dataset.

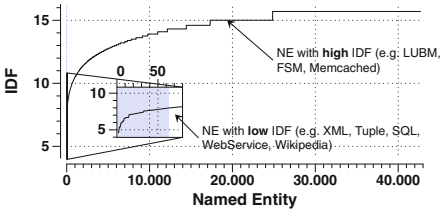

Fig. 3. Distribution of IDF values of extracted named entities.

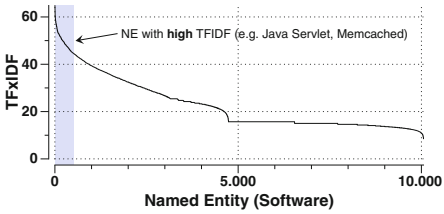

Fig. 4. Distribution of TFIDF values for NEs contained in *software* sentences.

Only a handful of named entities (about 100) feature a low IDF values (indicating that they are likely not fitting their assigned class well), while a large amount of entities (more than 60%) have relatively high informativeness. But, what is the quality of such entities? Are they useful in the characterization of class-specific sentences? To answer these questions, we first calculated a class-specific TFxIDF value for each named entity NE_i in the dataset as $TFIDF(NE_i, C_j) = (1 + log(|NE_{i,j}|)) \times IDF_{NE_i}$, where $|NE_{i,j}|$ is the raw frequency of a named entity NE_i within the sentences classified as relate to the class C_j. Then, for each class, we ranked named entities in decreasing order of $TFIDF(NE_i, C_j)$, and manually analyzed the first 100 entities.

Figure 4 shows an example distribution of TFIDF values. We excluded from this analysis the *objective* class, as objectives are usually not represented well by a single named entity, but instead require a more elaborate verbal description (which is usually fittingly provided by a rhetorical mention).

Table 6 shows examples of relevant named entities for each considered class. In terms of retrieval precision, we can observe promising results. NEs contained in *method* and *software* sentences feature a precision of 72% and 64%, respectively. On the other hand, NEs contained in *dataset* and *results* sentences resulted in a precision of 23% and 22%. In both cases, however, the returned entities are still relevant and related to the class: False positives in *dataset* sentences are mainly due to terms that are clearly related to data (e.g. Fuzzy set, Data model, Relational Algebra), but not specifically referring to actual datasets. Likewise, false positives in *results* sentences are mainly due to the presence of acronyms

Table 6. Examples of representative Named Entities in different classes

Dataset	Method	Software	Result
MovieLens	Collaborative Filtering	Java Servlet	Expected Value
Enron	Dynamic Programming	Portlet	Standard Deviation
IMDb	Active Learning	PHP	Precision and Recall
YAGO	Support Vector Machine	Memcached	P-value
DBPedia	Language Model	DOM API	MRR

that could be linked to the names of the methods tested in the paper. This type of error can be attributed the sentence-level granularity of our rhetorical mention detection, and can likely be reduced by including a boundary classifier into our workflow.

In summary, we can conclude that our approach is indeed suitable for extracting entities with respect to the five DMS classes in a meaningful and descriptive fashion. However, there are still some false positives of related concepts which cannot easily be recognized using simple statistic means, and which thus invite further deeper semantic filtering in future works.

6 Conclusion

In this paper, we presented the design and evaluation of knowledge extraction workflow aimed at extracting semantically rich metadata from scientific publications. The workflows specialises on the extraction of information related to data processing pipelines, with a focus on rhetorical mentions related to datasets, methods, software, objectives, and results. The extracted information is collected and published as a RDF knowledge base according to the `DMS` (Data Method Software) ontology, which was specifically designed to enable the description and linking of information related to data processing pipelines. The generated metatada allows researchers and practitioners to access and discover valuable information related to the properties and limitation of data sources and data processing pipelines, based on current literature.

Differently from previous work, our workflow relies on a lightweight distant supervision approach, which features lower training costs (compared to traditional supervised learning) and acceptable performance. These properties make the approach suitable for reuse in additional knowledge domains related to scientific publication. We show that, despite its simple design, it is possible to achieve high precision and recall for all classes. From these classified sentences, we extracted (rather noisy) named entities, which we subsequently filtered and ranked, to select entities which promise high descriptive power for their class.

While promising, the obtained results suggest ample space for future improvements. For instance, it will be interesting to investigate the performance of more complex machine learning classifiers working on richer feature sets (e.g., word-embeddings, POS-tags, parse trees, etc.). Furthermore, for labelling rhetorical

mentions, our current granularity is on sentence level. This introduces some additional confusion when extracting named entities in cases that a sentence has multiple labels, or only parts of a sentence refer to a rhetorical mention while others do not. This limitation could be remedied by additionally training boundary classifiers, which can narrow down rhetorical mentions more precisely. Furthermore, we employ sample filtering of entities based on statistics. This could be improved by further utilising semantic information from open knowledge bases.

Finally, we will address the application of our approach to real-life use cases. For instance, applications in the domain of digital libraries seem promising, allowing for both more meaningful queries to find relevant publications, and also allowing for analytic capabilities to track and visualise trends and changes in research fields over time.

References

1. Alexandru, C., Peroni, S., Pettifer, S., Shotton, D., Vitali, F.: The document components ontology (DoCO). Semant. Web **7**(2), 167–181 (2016)
2. Möller, K., Heath, T., Handschuh, S., Domingue, J.: Recipes for semantic web dog food — The ESWC and ISWC metadata projects. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 802–815. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_58
3. Glaser, H., Millard, I.: Knowledge-enabled research support: RKBExplorer.com. In: Proceedings of Web Science, Athens, Greece (2009)
4. Ghavimi, B., Mayr, P., Vahdati, S., Lange, C.: Identifying and improving dataset references in social sciences full texts. arXiv preprint arXiv:1603.01774 (2016)
5. O'Seaghdha, D., Teufel, S.: Unsupervised learning of rhetorical structure with untopic models. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014) (2014)
6. Tuarob, S., et al.: AlgorithmSeer: a system for extracting and searching for algorithms in scholarly big data. IEEE Trans. Big Data **2**(1), 3–17 (2016)
7. Osborne, F., Ribaupierre, H., Motta, E.: TechMiner: extracting technologies from academic publications. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 463–479. Springer, Cham (2016). doi:10.1007/978-3-319-49004-5_30
8. Khodra, M.L., et al.: Information extraction from scientific paper using rhetorical classifier. In: International Conference on Electrical Engineering and Informatics (ICEEI) (2011)
9. Helen, A., Purwarianti, A., Widyantoro, D.H.: Rhetorical sentences classification based on section class and title of paper for experimental technical papers. J. ICT Res. Appl. **9**(3), 288–310 (2015)
10. Burns, G.A., Dasigi, P., de Waard, A., Hovy, E.H.: Automated detection of discourse segment and experimental types from the text of cancer pathway results sections. Database. J. Biol. Databases Curation (2016)
11. Sateli, B., Witte, R.: What's in this paper? Combining rhetorical entities with linked open data for semantic literature querying. In: Proceedings of the 24th International Conference on World Wide Web. ACM (2015)
12. Liakata, M., Teufel, S., Siddharthan, A., Batchelor, C.R.: Corpora for the conceptualisation and zoning of scientific papers. In: LREC (2010)

13. Gil, Y., Ratnakar, V., Garijo, D.: Ontosoft: capturing scientific software metadata. In: International Conference on Knowledge Capture, p. 32. ACM (2015)
14. Groza, T.: Using typed dependencies to study and recognise conceptualisation zones in biomedical literature. PloS One **8**(11), e79570 (2013)
15. Dorgeloh, H., Wanner, A.: Formulaic argumentation in scientific discourse. In: Corrigan, R., Moravcsik, E.A., Ouli, H., Wheatley, K.M. (eds.) Formulaic Language, vol. 2, pp. 523–544. John Benjamins, Amsterdam (2009)
16. English for Writing Research Papers Useful Phrases. http://www.springer.com/cda/content/document/cda_downloaddocument/Free+Download+-+Useful+Phrases.pdf?SGWID=0-0-45-1543172-p177775190
17. Mesbah, S., Bozzon, A., Lofi, C., Houben, G.-J.: Describing data processing pipelines in scientific publications for big data injection. In: WSDM Workshop on Scholary Web Mining (SWM), Cambridge, UK (2017)
18. Lopez, P.: GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications. In: Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) ECDL 2009. LNCS, vol. 5714, pp. 473–474. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04346-8_62
19. Lipinski, M., Yao, K., Breitinger, C., Beel, J., Gipp, B.: Evaluation of header metadata extraction approaches and tools for scientific PDF documents. In: JCDL, Indianapolis, USA (2013)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: International Joint Conference on Natural Language Processing of the AFNLP, Singapore (2009)