Ontology Driven Extraction of Research Processes

Vayianos Pertsas[1(✉)], Panos Constantopoulos[1,2], and Ion Androutsopoulos[1,2]

[1] Department of Informatics, Athens University of Economics and Business, Athens, Greece
{vpertsas, panosc, ion}@aueb.gr
[2] Digital Curation Unit, IMSI - Athena Research Centre, Athens, Greece

Abstract. We address the automatic extraction from publications of two key concepts for representing research processes: the concept of research activity and the sequence relation between successive activities. These representations are driven by the Scholarly Ontology, specifically conceived for documenting research processes. Unlike usual named entity recognition and relation extraction tasks, we are facing textual descriptions of activities of widely variable length, while pairs of successive activities often span multiple sentences. We developed and experimented with several sliding window classifiers using Logistic Regression, SVMs, and Random Forests, as well as a two-stage pipeline classifier. Our classifiers employ task-specific features, as well as word, part-of-speech and dependency embeddings, engineered to exploit distinctive traits of research publications written in English. The extracted activities and sequences are associated with other relevant information from publication metadata and stored as RDF triples in a knowledge base. Evaluation on datasets from three disciplines, Digital Humanities, Bioinformatics, and Medicine, shows very promising performance.

Keywords: Ontology population · Information extraction
Machine learning methodologies · Linked data

1 Introduction

The steep increase of scientific publications in every major discipline [1] makes it increasingly difficult for experts to maintain an overview of their domain, increases the risk of missing new work or reinventing solutions, and makes it harder to relate ideas from different domains [2]. This situation could be significantly alleviated by supporting queries such as: find all papers that address a given problem; how was the problem solved; which methods are employed by whom in addressing particular tasks; etc. Answering queries like these essentially requires access to information about research processes. Such information could be compiled interactively, or automatically extracted from research publications, finally offered in a structured form suitable for

D. Vrandečić et al. (Eds.): ISWC 2018, LNCS 11136, pp. 162–178, 2018.
https://doi.org/10.1007/978-3-030-00671-6_10

supporting semantic queries. It is to be noted that search engines widely used by researchers, such as Google Scholar[1], Scopus[2] or Semantic Scholar[3], mostly leverage article metadata, while knowledge expressed in the actual text is only exploited in a shallow manner mostly by matching query terms to documents [3].

Understanding and encoding the knowledge contained in research articles is a complex task which poses several challenges. For instance, in order to extract the context of the research reported in an article (who is involved, what are their interests, affiliations, etc.), information from the metadata of the article must be extracted, analyzed and mapped onto a schema, so that activities, entities etc. extracted from the text of the article can be placed in the right context. Furthermore, the actual text of publications needs to be processed in order for activities, entities, and more generally concepts relevant to the documentation of research processes to be identified, extracted and associated according to predefined relation types of the same schema.

In this paper we address the problem of automatically extracting from publications, in the English language, two key concepts for representing research processes: the concept of *research activity* and the *sequence relation* between successive activities. We associate the information extracted from the texts of the articles with relevant information previously extracted from the articles' metadata or other digital repositories, and publish the resulting information in the form of RDF triples adhering to Linked Data standards. We consider these to be the first steps towards populating an ontology specifically designed for modeling research processes and practices [4], thus generating a research process documentation knowledge base.

Research activities and sequence relations manifest themselves in texts in ways that need to be specifically taken into account in order to achieve satisfactory extraction performance. For example, unlike usual named entities (e.g., persons, locations), research activities have textual descriptions of widely variable length, while pairs of successive (in time) activities often span multiple sentences, unlike simpler relation extraction tasks. We engineered several task-specific features exploiting the semantic context of the ontology being populated, syntactic dependencies of words and other syntactic structure information, which we combined with word embeddings. The latter are dense vector representations of words that can be produced in an unsupervised manner from unlabeled corpora and have proved instrumental in many Natural Language Processing (NLP) tasks in the past years [5, 6]. We actually employ three kinds of embeddings: word embeddings, part-of-speech (POS) tag embeddings, and dependency embeddings, all pre-trained for the domain of research processes, following the example of [7] where the first two kinds were combined.

We developed and compared several sliding window classifiers[4], thus exploring the activity and sequence extraction tasks along three dimensions:

(1) *Processing granularity*. We tested the effectiveness of classification at three levels of granularity: token-, sentence- and chunk-based classification.

[1] https://scholar.google.com/.

[2] https://www.elsevier.com/solutions/scopus.

[3] https://www.semanticscholar.org.

[4] Our software and data will be available at: http://nemo.dcu.gr/resources/.

(2) *Feature space*. The usual NLP practices were extended with the special task-specific features we developed and we assessed their effectiveness.
(3) *Machine learning (ML) method*. We developed classifiers employing Logistic Regression (LR) [8], linear Support Vector Machines (SVM) [9], and Random Forests (RF) [10], as well as a two-stage pipeline combination.

The performance of these classifiers was evaluated with datasets from three different disciplines: Digital Humanities, Bioinformatics, and Medicine. We measured Precision, Recall and F1 scores in token- and entity-based evaluations with very promising results, indicating the potential for creating a reliable research process knowledge base. The results also confirmed the contribution of the specially designed features in achieving that performance. We view the methods presented in this paper as strong baselines for extending our work to extracting other entities and relations describing research processes (e.g. goals, methods employed, propositions, etc.), and for experimenting with other classifiers (e.g., CRFs [11]), especially deep learning-based ones (e.g., RNNs, CNNs [12]) when larger datasets become available.

The rest of this paper proceeds as follows: in Sect. 2 we present related work and explain how our task is different; in Sect. 3 we describe the methodology and experimental setup; in Sect. 4 we discuss the evaluation experiments and their results; and we conclude in Sect. 5.

2 Related Work

To the best of our knowledge, the task of extracting variable-length textual descriptions of research activities from publications, and associating them on the basis of sequential order as inferred from the text, has not been addressed in previous work. That said, however, information extraction (IE) from scientific papers has attracted a lot of interest over the past years, as testified by the recent creation of a challenge on Scientific Information Extraction (ScienceIE) [3], the ACL RD-TEC Reference Dataset for Terminology Extraction and Classification [13], or domain-specific competitions such as BioCreAtIve[5]. Recent works deal with the extraction of key-phrases denoting tasks, scientific methods and materials from research documents [14, 15], the association of the extracted entities with Linked Data [16–18], or the recognition of biomedical entities such as genes [19, 20]. They use features based on surface form, POS tags, or word embeddings and they employ classifiers such as SVMs, CRFs or neural networks, to extract key-phrases and named entities from text, as well as binary lexical semantic relations (synonym-of, hyponym-of).

In [21], key-phrases denoting the “Focus”, “Technique” and “Domain” of the articles are identified on the basis of syntactic patterns matched via rules to the dependency tree of each sentence in article abstracts. In [22], rule-based methods are employed in understanding the dynamics preceding the creation of new topics. In [23], sentences from abstracts in the domains of clinical trials and biomedicine are classified

[5] http://biocreative.sourceforge.net/index.html.

in categories, such as introduction, purpose, method, results and conclusion, using various bag-of-words or bag-of-n-grams representations.

A specialized system for extracting specific elements from legal contracts [7] uses sliding window classifiers and handcrafted features combined with word and POS tag embeddings to extract contract elements such as title, date, signatories' names, etc.

In [24, 25], portions of text mentioning specific papers are extracted and relations to the corresponding citations are generated using rule based approaches or features that deal mainly with the surface form or structural aspects of text (e.g., they examine the existence of specific POS tags or lexical terms that indicate references, other citations, etc. in the current or previous sentence). In [26], authors and organizations are identified in scientific papers via CRFs using features that mainly deal with token surface form (lower/upper case, presence in gazetteers, font size, etc.) or structural text characteristics (appearance in sections/paragraphs, first word in line, etc.). The extracted entities are then interrelated by further extracting the *hasAffiliation* property. For that, an SVM with Gaussian kernel is used with features related to the author affiliation markers and the distance of extracted strings.

In other works related to action sequencing, such as [27], the authors create abstractions of action sentences based on a predefined template and then cluster those abstractions together based on a functional similarity measure. In [28], the authors use deep reinforcement learning, in order to extract sequences of labeled actions from sentences; each action is represented by arguments constructed from the verb and its object (e.g., cook (rice)) and the sequencing relations can be selected or eliminated based on their type (i.e., optional, exclusive or essential). In [29], the authors use a predefined list of names to map their action descriptions and interpret them as action sequences, or to generate navigational action descriptions using an encoder-aligner-decoder structure. Unlike the above methods, we identify and associate actions that are not expressed by single words or mapped to a fixed template or list of names. Instead, in our work actions have complex textual representations of variable length and cannot be labeled with words from a name-list. Moreover, we are not confined to deriving sequence relations from single lexical keywords. Instead, sequence relations are inferred from a combination of the actual textual context of activities along with structural properties of the text (e.g., relative positions of the entities in the texts).

In all of the approaches reviewed above, IE from text is addressed using either rules or ML methods based on features that handle mainly the surface form of words disregarding other information, such as attributes derived from syntactic dependencies or more complex syntactic patterns. ML methods of that kind perform inadequately in extracting research activities from text, as suggested by the evaluation of our baseline method that uses similar features. This behavior can be attributed to the following characteristics of the task at hand:

- Research activities are entities manifesting themselves only by their textual description and not by any specific nomenclature. Furthermore, their textual description does not follow any specific surface form.
- The textual chunks representing research activities can be of arbitrary length. This has been observed to exceed 50 tokens, which is significantly higher than the lengths of entity names in common Named Entity Recognition (NER).

- Unlike other NER tasks, the surface form of the tokens inside the textual description of a research activity can vary so much, that it is insignificant for the purpose of extracting activities.
- Contrary to common NER tasks, where the extracted entities cover only a small portion of a sentence, research activities may cover almost entire sentences, and even more than one sentence. Here we restrict our investigation to activities contained in one sentence each. Multi-sentence activities are usually composed of smaller ones. The hierarchical decomposition of those composite activities eventually leads to simple single-sentence ones.
- Sequence relations between activities cannot be detected solely from lexical cues in the text. Other attributes of the activities, including their relative position in text, actual textual description, etc., are also employed to improve classification.

The main contributions of the work reported here are:

- The way we address the complexity of the particular task by combining information from the ontology (e.g., available relation types, constraints on their domain and range), task-specific embeddings of words, POS tags, and syntactic dependencies, features detecting special syntactic sequences of words and their order of appearance in texts, specialized features dealing with lexico-syntactic patterns, as opposed to just word surface form, currently employed in other works related to extracting knowledge from scientific literature.
- The proposed methods are applicable to any scientific domain, since no domain-specific lexica or training corpora are required, and they are demonstrated with test sets from three disciplines, capturing a variety of writing styles.
- Our methods yield higher performance compared to common NER or rule-based solutions, as evidenced by comparing to the baselines, which is notable especially considering the fact that the limited sizes of the datasets we had available do not allow for more sophisticated ML approaches (such as deep learning methods).

Furthermore, we show how –based on the semantics from an ontology, specifically designed to represent research processes [4] - information extracted from text can be associated with knowledge from article metadata and other sources (such as ORCID[6]) as part of creating a comprehensive research process knowledge base.

3 Setup and Methodology

We use as schema for research process knowledge bases the Scholarly Ontology (SO) [4], a domain-independent ontology of scholarly/scientific work. A specialization, in fact precursor, of SO already applied to the domain of Digital Humanities is the NeDiMAH Methods Ontology (NeMO) [30]. A brief overview of SO core concepts is given in the following section. For a full account see [4].

[6] https://orcid.org.

3.1 Conceptual Framework: The Scholarly Ontology

Figure 1 shows the core concepts and relations in SO. The rationale behind the ontology is to support documenting "*who* does *what*, *when*, and *how*" in a given scholarly domain. The ontology is built around the central notion of *activity* and combines three perspectives: the *agency* perspective, concerning actors and intentionality; the *procedure* perspective, concerning the intellectual framework and organization of work; and the *resource* perspective, concerning the material and immaterial objects consumed, used or produced in the course of activities.

Activity concerns real events that have occurred in the form of intentional acts carried out by actors. The instances of the *Activity* class are real processes with specific results, as opposed to those of the *Method* class, which are specifications, procedures, or recipes for carrying out activities so as to address specific goals. Sequence and composition of activities are represented by the *follows* and *partOf* relations respectively. *Actor* instances are entities capable of performing intentional acts that they can be accounted or referenced for. They can participate in activities, actively or passively, in one or more roles. Subclasses of *Actor* are the classes *Person* and *Group*, representing individual persons and collective entities respectively. Further specializations of *Group* are the classes *Organization* and *ResearchTeam*. *ContentItem* comprises information resources, regardless of their physical carrier, in human readable form (with images, tables, articles, bibliographic references, etc. being specializations of *ContentItem* class). *Assertion* includes all kinds of assertions in the scholarly domain and captures the intellectual essence of scholarly activity, comprising propositions resulting from activities and can be *supportedBy* evidence provided by content items. Finally, the class *Topic* comprises thematic keywords which function as tags expressing the subject of methods, the topic of content items, research interests of actors, etc.

In this paper, we focus on extracting from text and automatically populating two key concepts of the ontology: (i) *Activity*, a unary predicate denoting research processes

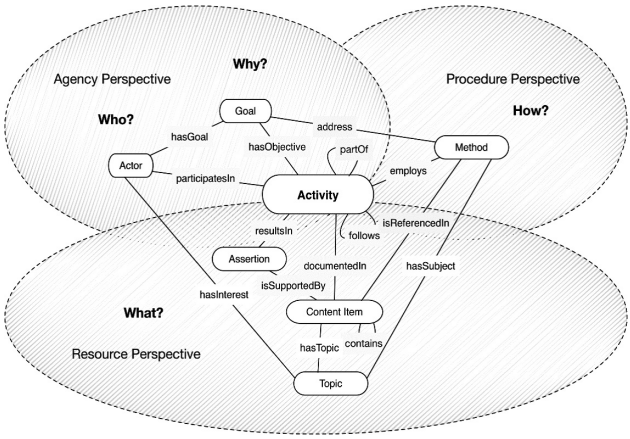

Fig. 1. Scholarly ontology core

such as a biological experiment, an archeological excavation, an anthropological or medical study, etc. and (ii) *follows*, a binary predicate denoting the sequence relation between two successive activities. Figure 2 shows an example of textual chunks representing research activities -highlighted- and their sequence relations.

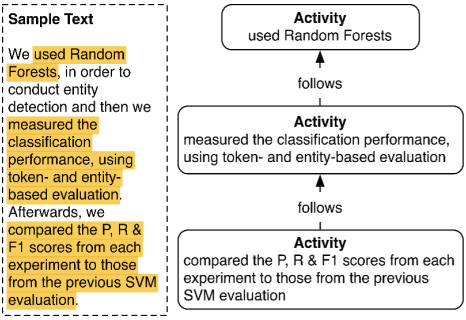

Fig. 2. Activities and sequential relations

3.2 The Dataset

An unlabeled dataset obtained from 50,000 open-access research papers was used in order to create embeddings. The dataset consisted of approximately 10,000,000 sentences after metadata cleaning and parsing using spaCy[7], yielding 300,000,000 tokens and eventually a vocabulary of approx. 1,000,000 unique words (types). Word, part-of-speech tag (POS) and dependency (DEP) embeddings were generated from the above. Specifically: 100-dimensional word embeddings were produced using the Gensim implementation of word2vec[8] (skip-gram model); 25-dimensional POS embeddings were produced by replacing each token by its corresponding POS tag before running word2vec; and 25-dimensional DEP embeddings were produced by replacing each token by the label of the (unique) arc linking the token to its head in the dependency tree. Our experiments with other general-purpose, publicly available embeddings, such as those trained on the Common Crawl corpus using GloVe[9], or those trained on Wikipedia articles with word2vec, showed inferior performance compared to our domain-specific embeddings. This can be attributed to the fact that our embeddings are trained exclusively on scholarly articles, thus capturing the idiosyncrasies of scholarly writing styles.

To train and evaluate our machine learning methods, we used research articles randomly selected using APIs from publishers such as Springer and Elsevier, or by scraping online journals such as the Digital Humanitites Quarterly. To annotate the

[7] https://spacy.io/.

[8] https://radimrehurek.com/gensim/.

[9] https://nlp.stanford.edu/projects/glove/.

dataset with ground truth, we used human annotators, appropriately trained in the use of SO. Guidelines and examples were provided to the annotators.

The training set, comprising texts from 50 research articles covering 9 research domains, was annotated by two post-graduate students. Three annotation trials (one article annotated by both annotators per trial, followed by discussion) were initially performed. Inter-annotator agreement was 81% kappa statistic, measured on 5 articles annotated by both annotators at the end of the annotation trials. Subsequently, the remaining articles were annotated by one annotator each. The annotation of the training set yielded approx. 1,000 sequence relations and 1,700 activities comprising approx. 31,000 tokens. For hyper-parameter tuning we used 3-fold cross-validation.

For testing, we used articles from three disciplines, Digital Humanities (DH), Bioinformatics (BIOINF) and Medicine (MED), to expose our classifiers, trained on a generic set, to a wide variety of writing styles. Three test sets, 15 articles per discipline, were annotated by two expert -per discipline- annotators. The annotators were trained on 5 articles per discipline, annotated by both annotators, with discussion after annotating each article. Inter-annotator agreement was 81%, 83% and 85% kappa for DH, BIOINF and MED, respectively, for the fifth article of each discipline. The remaining articles were annotated by one annotator each. For each test set, human annotation produced approx. 600 activities containing approx. 10,000 tokens. Concerning sequence relations, human annotation produced approx. 200 relations for DH, 500 for BIOINF, and 600 for MED. The differences in the numbers can be attributed to the granularity of activities and the writing style prevalent in each research field.

3.3 Extracting Research Activities

Seven sliding window classifiers (SWC) and a two-stage pipeline classifier were implemented for extracting research activities (Table 1). They all perform token-based classification by examining each token t and its surrounding tokens in a fixed-size window, and classifying t as positive if it is part of a phrase expressing a research activity, or negative otherwise. The size of the window was set at 30 tokens around t (a total of 30 + 30 + 1 = 61 tokens) following hyper-parameter tuning. Zero-padding was used to represent tokens exceeding the sentence boundary. Each window of tokens was turned into a feature vector representing the token t being classified. We experimented with Logistic Regression, linear Support Vector Machines and Random Forests, with different feature specifications as detailed below. We use the notation M.E.F or M.E.F.F to denote the resulting classifiers, where M denotes the learning method used, E the embeddings and F the special features.

The first and second classifiers, **LR.WP.B** and **SVM.WP.B**, use Logistic Recression (LR) and linear SVM respectively, while they both employ 139 features: 125 derived from the 100- and 25-dimensional vectors of the word and POS embeddings (WP), and another 14 binary hand-crafted features labeled "basic" (B) that deal with the surface form of tokens. Of those features, 7 capture specific token surface forms (title, capitalized, digit, punctuation mark, etc.), while the other 7 determine whether the token's lexical form indicates neighboring activities. For example, words that indicate sequencing of events ('first', 'afterwards', 'finally', etc.), specialization ('concretely',

'specifically', etc.), causality ('for', 'to', etc.,), etc. The total number of features in the window is: 61 × 139 = 8,479.

The third and fourth classifiers, **LR.WPD.BS** and **SVM.WPD.BS**, differ from the first two in that they extend the embeddings-related features with 25 originating from DEP embeddings (WPD) and the special features with 10 binary "smart" features (BS) related to special syntactic structures. The latter are meant to capture the inclusion of a token in patterns suggesting activities, either directly, such as sub-sentences with verb in past tense and subject in first person (e.g.: "we performed stylistic analysis"), or indirectly, such as sub-sentences with causal modifiers indicating goals of neighboring activities (e.g. "[ACT: performed stylistic analysis], in order [GOAL: to recognize each characteristic]"). The total number of features is now 61 × (139 + 25 + 10) = 10,614.

The fifth classifier, **RF.PD.BS**, employs Random Forests (RF) and uses 51 one-hot features representing POS tags and 71 one-hot features representing DEP tags (PD), rather than embeddings. It also uses the same binary features (14 "basic" and 10 "smart") as the third and fourth classifiers. The total number of features in the sliding window is 61 × (14 + 10 + 57 + 71) = 9,272.

The sixth and seventh classifiers, **LR.PD.S.BS** and **SVM.PD.S.BS,** are like the third and fourth with the difference that: (a) they omit features related to word embeddings, and (b) they account for the syntactic sequence (S) of words, i.e., the sequence from the syntactic dependency of the word to its head and the head of its head, thus encoding joint information for 3 tokens instead of just one. As an example of such a syntactic sequence, consider in the first sentence of Fig. 2, the word "conduct", with its syntactic head "order" and the syntactic head of its head "in". The total number of features in the sliding window is now 61 × (10 + 14 + 50) × 3 = 13,542.

In addition to the above classifiers we implemented a two-stage pipeline (see Fig. 3). The first classifier, **SVM.WPD.BS,** is trained on all the sentences of the training set, as before, but now performs sentence classification instead of token classification, i.e., detects only the existence of research activities in the sentence without identifying their boundaries. For the first classifier, each sentence is represented using averaged word/POS/DEP embeddings of the contained tokens. This produces a vector of 100 or 25 features derived from the 100-dimensional word embeddings or the 25-dimensional POS or DEP embeddings respectively, keeping the number of features/dimensions independent from the actual number of tokens in the sentence. In addition, we used 14 binary features for representing the existence or absence inside the sentence of the -previously described- special syntactic patterns and lexical forms that provide indirect activity identifiers. For the second classifier, we used **SVM.PD.S.BS**, but now trained only on sentences containing at least one research activity. This performs token-based classification and determines the boundaries of the chunks describing research activities in the sentences classified as positives by the first classifier. The intuition behind the pipeline is that, by splitting the task into two simpler sub-tasks, each separate classifier will achieve high enough accuracy for their concatenation to produce better results, which was proven correct in the evaluation.

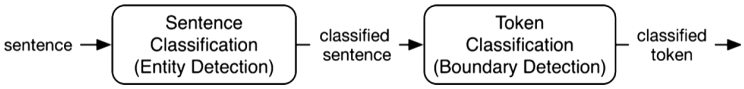

Fig. 3. Activity extraction pipeline

3.4 Extracting Sequence Relations

Extracting sequence relations requires examining all plausible activity pairs. For every pair of extracted activities, the text chunk bounded by these two entities, [*act1*, …, *act2*], is treated as expressing a candidate sequence relation. A maximum chunk length of 500 tokens, set during hyper-parameter tuning, serves to restrict the search to a reasonable set of candidates excluding pairs of too distant entities unlikely to be sequential, yet including pairs of entities from neighboring paragraphs or sections with reasonable chance of being related. A classifier then determines whether the bounding activities of the chunk satisfy the property *follows*.

Each chunk is represented using averaged word/POS/DEP embeddings of the tokens in the chunk together with 11 special features: 5 that examine certain structural properties of the chunks (*act1* and *act2* are in the same sentence/adjacent sentences/same paragraph; other entities intervene; the chunk contains conjuncts, like the word "and", syntactically associated with tokens inside the boundary entities); 3 for *act1* and 3 for *act2* that examine the entire sentence(s) containing each one of them in order to capture possible sequence indicators (e.g. the words "*then"* and "*Afterwards"* in Fig. 2) referring to *act1* and *act2*, even when they are not inside the chunk bounded by *act1* and *act2* or the individual chunks representing *act1* and *act2* respectively.

We implemented three classifiers for extracting sequence relations between activities. The first sequence extractor, **LR.WPD.B**, uses Logistic Regression and 161 features per chunk: 100 features for the averaged word embeddings of the tokens in the chunk, 25 for the averaged POS, 25 for the averaged DEP embeddings, 5 for structural chunk properties and 6 for sequence indicators, as discussed above. The second extractor, **SVM.WPD.B**, uses the same features, but with a linear SVM. The third extractor, **RF.PD.B**, uses Random Forests (RF) and the per-dimension sum of the one-hot encodings of the POS and DEP tags of each token in the chunk. We also experimented with the average and the TF-IDF-weighted average of the encodings, but without better results in either case.

3.5 Background Context Integration and URI Creation

Having extracted research activities and their sequence relations, we attach to them contextual information obtained from the metadata of the publications. Specifically, we have created mappings that currently support the association of article metadata from two major publishers (Springer and Elsevier) with relevant SO classes such as participants in the research processes (the authors of the paper), their interests (author keywords) and their personal information (affiliations, email, etc.), the *ContentItem* that they are documented in (the research articles), etc. We also provide integration through API with ORCID, a non-for-profit organization for assigning unique, persistent IDs to

researchers, so that (i) the ORCID id of each person can be used for duplicate detection and (ii) additional information regarding related projects, funding or biography can be retrieved through the ORCID repository.

The research process knowledge base is created by encoding the extracted information as RDF triples adhering to Linked Data principles and the RDFS[10] and NIF[11] models. For entities with a proper name, such as *Persons*, *Organizations*, *Articles* and *Topics*, their URIs are derived by combining the namespace of the knowledge base, the entity type according to SO, and a unique id provided by the entity name (such as ORCID id or email for persons, article id, topic name, etc.). For activities and sequence relations, URIs are generated by combining the namespace of the knowledge base, the entity type according to SO, the source of extraction (publication id) and the two offsets identifying the boundaries of the extracted entity inside the text, thus ensuring that each URI is unique. A small excerpt of the knowledge base is shown in Fig. 4. Based on our measurements, information extracted from 50 articles translates roughly to 100,000 triples, this being highly dependent on the writing style and the discipline. Indicative running times (on a PC with an Intel i7, 16 GB RAM) for the entire process are approx. 100 s/article.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:ns1="http://dcu.gr/ontologies/so/instances/"
   xmlns:ns2="http://dcu.gr/ontologies/so/"
   xmlns:ns3="http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
  <rdf:Description rdf:about="http://dcu.gr/ontologies/so/instances/ContentItem-S1873506111000596/Activity#offset_13533_13533">
    <ns1:isDocumentedIn rdf:resource="http://dcu.gr/ontologies/so/instances/ContentItem-Elsevier-S1873506111000596"/>
    <ns3:referenceContext>http://api.elsevier.com/content/article/pii/S1873506111000596</ns3:referenceContext>
    <rdf:type rdf:resource="http://dcu.gr/ontologies/so/Activity"/>
    <ns1:hasParticipant rdf:resource="http://dcu.gr/ontologies/so/instances/Person-ORCID-0000-0001-9512-4708"/>
    <ns2:follows rdf:resource="http://dcu.gr/ontologies/so/instances/ContentItem-S1873506111000596/Activity#offset_13262_13262"/>
    <ns3:beginIndex>13533</ns3:beginIndex>
    <ns3:endIndex>13533</ns3:endIndex>
    <rdfs:label>performed token- and entity-based evaluation</rdfs:label>
  </rdf:Description>
```

Fig. 4. Excerpt from the produced RDF triples

4 Evaluation

In general, metadata association has exhibited very good performance since it relies solely on pre-constructed mappings between fixed schemas. Few isolated incidents (lower than 1%) of improper association were due to errors in XML/HTML tags in the article (e.g., an empty or misplaced bracket) and can be treated with additional escape rules as part of the general debugging process.

Regarding the information extraction from text, we evaluated the performance of all the classifiers by measuring Precision, Recall and F1 scores. After window-size selection and hyper-parameter tuning using 3-fold cross-validation on the training set, all the classifiers were trained on the entire training set. As previously mentioned, we

[10] https://www.w3.org/TR/rdf-schema/.

[11] http://persistence.uni-leipzig.org/nlp2rdf/.

used three different test sets, DH, BIOINF, and MED, presumably representing different writing styles, as well as their combination (ALL Test Set).

Approximate Randomization Tests (ART) [32] between every classifier and the relevant baseline were carried out to ensure the statistical significance of the tests. Classifiers were grouped in zones of statistically similar results (shown by dividing lines in Tables 1 and 3) and ARTs were run on every combination of methods from different zones in order to ensure that the difference between any two measurements is statistically significant given our test sets. The Bonferroni correction was used to adjust the threshold (p-value) from the default 0.05 to 0.00625 for activity extraction and 0.0125 for sequence relation extraction, since we compared more than two systems. All pair combinations gave probabilities below the above thresholds in ARTs, therefore all the results shown are statistically significant.

4.1 Research Activity Extraction Evaluation

The evaluation of activity extraction methods involves comparing classifier results against a reference standard produced by human annotators on the basis of Precision, Recall and F1 scores calculated as usual[12]. In addition, we compare the classifiers with a "baseline" method, similar to those commonly used in NER tasks [7], with a smaller sliding window of 15 tokens (7 left, the central token *t*, 7 right), 100 features for word embeddings, 25 features for POS embeddings and 14 "basic" binary features for surface form representation, in total $15 \times (125 + 14) = 2{,}085$ features. The baseline uses a linear SVM trained on the same training set, as this has proved experimentally to perform slightly better than LR and RF. Two groups of comparisons are made: token-based and entity-based.

In token-based evaluation, a *true positive* (TP) is a token correctly classified as part of a chunk representing a research activity, a *false positive* (FP) is a token incorrectly classified as part of a research activity, and a *false negative* (FN) is a token incorrectly classified as non-part of a research activity. Results of the token-based evaluation for each test set are shown in Table 1. Regarding the pipeline classifier which consists of a sentence- and a token-based classifier in tandem, detailed per stage and aggregate performance results are shown in Table 2. The aggregate scores of the pipeline are also shown in Table 1 for comparison with the other methods.

The **Pipeline** classifier achieved the highest scores on every test set and criterion. The aggregate performance of the pipeline is inferior to that of the individual stages (see Table 2) due to error propagation, since the sentences that are wrongly classified in the first classifier are fed as input into the second. The baseline, on the other hand, performed worse than all the other classifiers on every test set and criterion. This can mainly be attributed to two factors: (a) the difference in the size of the sliding window (as indicated from the performance increase between the baseline and the **SVM.WP.B**); and (b) the use of the DEP embeddings and the "smart" features. Moreover, word embeddings do not add much to the overall improvement of the classification, as suggested by the performance of the **RF.PD.BS**, **LR.PD.S.BS** and **SVM.PD.S.BS**

[12] $P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F1 = \frac{2*P*R}{P+R}$.

Table 1. Token-based evaluation

		DH test set			BIOINF test set			MED test set			ALL test set		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
	Baseline	0.54	0.30	0.38	0.76	0.50	0.60	0.76	0.62	0.69	0.72	0.50	0.59
1	LR.WP.B	0.62	0.44	0.52	0.79	0.59	0.68	0.79	0.66	0.72	0.75	0.58	0.65
2	SVM.WP.B	0.60	0.50	0.54	0.80	0.66	0.72	0.78	0.68	0.73	0.74	0.63	0.68
3	LR.WPD.BS	0.78	0.76	0.77	0.83	0.81	0.82	0.88	0.83	0.85	0.84	0.80	0.82
4	SVM.WPD.BS	0.76	0.80	0.78	0.83	0.83	0.83	0.87	0.85	0.86	0.83	0.83	0.83
5	RF.PD.BS	0.79	0.80	0.80	0.85	0.83	0.84	0.89	0.83	0.86	0.85	0.82	0.83
6	LR.PD.S.BS	0.77	0.79	0.78	0.82	0.83	0.83	0.88	0.88	0.88	0.83	0.84	0.84
7	SVM.PD.S.BS	0.79	0.82	0.80	0.84	0.84	0.84	0.89	0.89	0.89	0.85	0.85	0.85
8	SVM-Pipeline	**0.83**	**0.82**	**0.82**	**0.87**	**0.89**	**0.88**	**0.90**	**0.93**	**0.92**	**0.87**	**0.89**	**0.88**

Table 2. Pipeline evaluation

	DH test set			BIOINF test set			MED test set			ALL test set		
Entity identification:	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SVM.WPD.BS	0.90	0.89	0.89	0.96	0.94	0.95	0.96	0.96	0.96	0.94	0.93	0.94
Boundary detection:												
SVM.PD.S.BS	0.92	0.89	0.90	0.92	0.95	0.94	0.95	0.96	0.95	0.93	0.94	0.93
Pipeline:												
SVM-Pipeline	0.83	0.82	0.82	0.87	0.89	0.88	0.90	0.93	0.92	0.87	0.89	0.88

classifiers, as word embeddings can be replaced by other contextual information regarding the syntactic sequence of tokens. Therefore, the distinctive features of the methods we developed prove to contribute significantly to the performance of research activity extraction.

In entity-based evaluation, each maximal sequence of consecutive positive tokens is considered as a research activity ("entity"). Ideally, an entity is correctly predicted by a classifier only if it matches 100% with one annotated by humans, counting as errors even the slightest deviations. In practice, a close match suffices, especially in cases where the extracted entities are very long. A threshold of 86% was automatically selected by averaging the Levenshtein distances of a sample of 100 pairs of overlapping strings (a predicted and a gold entity in each pair) for which the annotators indicated that the overlap was sufficient. This translated roughly into a difference of 1-5 tokens (including punctuation marks) at the boundaries of each entity. Consequently, in entity-based evaluation a true positive (TP) is a predicted string that matches a reference standard string by at least 86%; a false positive (FP) is an un-matched predicted string; and a false negative (FN) is an un-matched reference standard string. Results of the entity-based evaluation are shown in Table 3.

The **RF.PD.BS** and **Pipeline** classifiers compete for best performance in the case of entity-based evaluation with similar results on most test sets. The baseline again performs worse than all other methods. Performance results in entity-based evaluation are

Table 3. Entity-based evaluation

		DH test set			BIOINF test set			MED test set			ALL test set		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
	Baseline	0.16	0.30	0.20	0.23	0.60	0.34	0.28	0.76	0.40	0.23	0.60	0.33
1	LR.WP.B	0.48	0.60	0.53	0.56	0.72	0.63	0.52	0.74	0.62	0.53	0.70	0.60
2	SVM.WP.B	0.42	0.64	0.50	0.54	0.76	0.63	0.51	0.78	0.62	0.50	0.74	0.60
3	LR.WPD.BS	0.58	0.80	0.67	0.57	0.80	0.66	0.58	0.82	0.68	0.58	0.80	0.67
4	SVM.WPD.BS	0.54	0.82	0.65	0.55	0.79	0.65	0.57	0.83	0.67	0.55	0.81	0.66
5	LR.PD.S.BS	0.59	0.82	0.69	0.58	0.78	0.66	0.60	0.84	0.70	0.59	0.81	0.68
6	SVM.PD.S.BS	0.61	0.83	0.70	0.62	0.76	0.68	0.61	0.83	0.70	0.61	0.80	0.70
7	RF.PD.BS	**0.68**	0.79	**0.73**	**0.66**	0.78	**0.72**	**0.66**	0.83	**0.74**	**0.67**	0.80	**0.73**
8	SVM-Pipeline	0.64	**0.83**	**0.73**	0.62	**0.84**	**0.72**	0.60	**0.86**	0.71	0.62	**0.85**	0.72

inferior to those in token-based evaluation. Error analysis showed that this can be attributed to tokens occurring in entity chunks incorrectly classified as not being research activities; this causes the split of the original entity into smaller ones, in turn producing additional errors (1 FN for the undetected original entity and 1 FP for each smaller entity). Consider, for instance, the second sentence in Fig. 2. Had the classifier produced 0 for the token "to" inside the sentence, the activity "compared the P, R and F1 scores from the previous experiment to those from the SVM evaluation" would have been split into two smaller entities: "compared the P, R and F1 scores from the previous experiment" and "those from the SVM evaluation". Since each of the new smaller entities matches the original by less than 86%, the resulting misclassification would give 2 FPs for the smaller activities and 1 FN for the original.

The performance decrease in entity evaluation was found to vary among domains. Indeed, a 6.9% average decrease in F1 scores was observed with the DH test set, while the decrease was 14.5% with the BIOINF test set, and 15.9% with MED. Error analysis indicates that this can be attributed mainly to the differences in writing style. For example, in the DH test set, the research activity entities were found to have smaller size and contain fewer "error prone" tokens (such as acronyms or formulas) that could cause individual token misclassification and thus split of the entity chunk.

4.2 Sequence Relation Extraction Evaluation

The evaluation of sequence relation extraction methods involves comparing the predicted relations among the reference standard entities in each test set with those produced by the human annotators on the basis of Precision, Recall and F1 scores calculated as usual. A true positive (TP) is a chunk [act1, …, act2] for which the classifier correctly predicted the follows (act2, act1) property; a false positive (FP) is a chunk for which follows (act2, act1) was incorrectly predicted; and a false negative (FN) is a chunk for which follows (act2, act1) incorrectly failed to be predicted. Classifier performance is also compared with that of a simple baseline method that assigns a sequence relation to all adjacent activities in a paragraph and activities connected by sequence cue words (e.g., "then", "subsequently"). Results are shown in Table 4.

Table 4. Relation extraction evaluation

		DH test set			BIOINF test set			MED test set			ALL test set		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline		0.62	0.72	0.67	0.65	0.89	0.76	0.59	0.92	0.72	0.62	0.88	0.72
1	LR(WPD)E-AVG-B	**0.87**	0.90	**0.88**	0.85	0.58	0.69	**0.94**	0.69	0.80	0.87	0.77	0.82
2	SVM(WPD)E-AVG-B	0.80	0.93	0.86	0.83	0.65	0.73	0.91	0.75	0.82	0.84	0.80	0.84
3	RF(PD)1H-SUM-B	0.81	**0.93**	0.87	**0.87**	**0.85**	**0.86**	**0.94**	**0.90**	**0.92**	**0.88**	**0.89**	**0.89**

In sequence relation extraction, **RF.PD.B** performed best in BIOINF, MED and overall (F1: 0.86, 0.92 and 0.89 respectively), while for the DH test set the forerunner was **LR.WPD.B** (F1: 0.88). Error analysis suggests that misclassifications are mostly due to adjacent sentences containing multiple activities, a situation more frequent in DH and BIOINF. For example, consider the excerpt: "[*act1:* Two-thirds of the extracted bootstrap samples were used for constructing the model] and then [*act2:* the other one-third were used for testing]. To calculate variable importance, we first [*act3:* put down the out-of-bag cases] and [*act4:* counted the number of votes cast for the correct class], and then [*act5:* randomly permuted the values of variable *root j* in the out-of-bag cases]". One classifier associated *act3* and *act4* of the second sentence with the last activity of the first sentence (*act2*), and another associated the first entity of the second sentence (*act3*) with each entity in the first (*act1, act2*). These predicted associations are treated as wrong because, by definition, *follows* only holds for *immediately successive* activities, with no others in between. Classifiers also tended to fail to detect activity sequences in texts where activities were sparse (e.g., no adjacent paragraphs with at least one activity each), probably because of the large size of text between activities and the structure (not adjacent paragraphs).

5 Conclusion

We addressed the automatic extraction from the text of publications of two core elements of research processes, research activities and their sequence relations, as a basic step towards populating research process knowledge bases complying to an ontology for research process documentation, the Scholarly Ontology (SO). We showed that the complexity of the task demands more complex feature engineering than usual NER tasks. We implemented and tested several sliding window classifiers employing features specifically designed to deal with particular lexical, syntactic, structural and semantic aspects of textual context. Alternative implementations were compared using linear SVMs, Logistic Regression, and Random Forests, as well as a two-stage pipeline classifier specifically configured for the task of activity extraction.

The classifiers were evaluated against a reference standard produced by human annotators, with three different test sets from three domains (Digital Humanities, Bioinformatics and Medicine) and very promising results: overall F1 score 0.88 for research activity extraction in token-based evaluation and 0.73 in entity-based evaluation, and 0.89 for sequence relation extraction. The classifiers were also compared

with simpler baselines which were configured without the special features of this work and with smaller sliding window size closer to those used in common NER tasks. The baseline classifiers were consistently inferior in both activity and sequence relation extraction, an additional evidence in support of the effectiveness of the special features and window width we employed. We also showed how contextual information from article metadata and other sources such as ORCID can be associated with the extracted entities according to the Scholarly Ontology and stored as RDF triples adhering to Linked Data standards.

Future work includes extracting further concepts for documenting research processes according to the Scholarly Ontology, such as goals, research questions, propositions, methods, etc., along with their corresponding relations (such as *partOf, employs, hasObjective*, etc.) and experimenting with more complex classifiers (e.g. CNNs or RNNs [12]) when additional larger training datasets become available.

References

1. Bornmann, L., Mutz, R.: Growth rates of modern science: a bibliometric analysis based on the number of publications. J. Assoc. Inf. Sci. Technol. Technol. **66**, 2215–2222 (2015)
2. Renear, A.H., Palmer, C.L.: Strategic reading, ontologies, and the future of scientific publishing. Science **325**, 828–832 (2009)
3. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: SemEval 2017 Task 10: ScienceIE, pp. 546–555 (2017)
4. Pertsas, V., Constantopoulos, P.: Scholarly ontology: modelling scholarly practices. Int. J. Digit. Libraries. **18**, 173–190 (2017)
5. Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: CoNLL, pp. 171–180 (2014)
6. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Trans. ACL **3**, 211–225 (2015)
7. Chalkidis, I., Michos, A., Androutsopoulos, I.: Extracting contract elements. In: ICAIL, pp. 19–28, London (2017)
8. McCullagh, P., Nelder, J.A.: Generalized Linear Models, Chapman and Hall London – New York (1983). 261 S
9. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000). ISBN 0-521-78019-5
10. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
11. Lafferty, J., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001. vol. 8, pp. 282–289 (2001)
12. Goldberg, Y.: A primer on neural network models for natural language processing. J. Artif. Intell. Res. **57**, 345–420 (2016)
13. QasemiZadeh, B., Schumann, A.-K.: The ACL RD-TEC 2.0: a language resource for evaluating term extraction and entity recognition methods. In: LREC, pp. 1862–1868 (2016)
14. Lee, L.-H., Lee, K.-C., Tseng, Y.-H.: The NTNU System at SemEval-2017 Task 10: extracting keyphrases and relations from scientific publications using multiple CRFs. In: 11th International Workshop on SemEval-2017, pp. 950–954 (2017)
15. Luan, Y., Ostendorf, M., Hajishirzi, H.: Scientific Information Extraction with Semi-supervised Neural Tagging, pp. 2631–2641. arXiv:1708.06075 (2017)

16. Sateli, B., Witte, R.: What's in this paper? Combining rhetorical entities with linked open data for semantic literature querying. In: ICWWW ACM, pp. 1023–1028 (2015). https://doi.org/10.1145/2740908.2742022
17. Osborne, F., de 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). https://doi.org/10.1007/978-3-319-49004-5_30
18. Sateli, B., Witte, R.: Semantic representation of scientific literature: bringing claims, contributions and named entities onto the Linked Open Data cloud. PeerJ Comput. Sci. **1**, e37 (2015)
19. Song, Y., Yi, E., Kim, E., Lee, G.G., Park, S.J.: POSBIOTM-NER: a machine learning approach for bio-named entity recognition (2004). Doi=10.1.1.101.1165
20. Plake, C., et al.: A support vector classifier for gene name recognition. In: BioCreAtIvE Workshop, Granada, Spain, pp. 1–5 (2004)
21. Gupta, S., Manning, C.: Analyzing the dynamics of research by extracting key aspects of scientific papers. In: IJCNLP, pp. 1–9 (2011)
22. Salatino, A.A., Osborne, F., Motta, E.: How are topics born? Understanding the research dynamics preceding the emergence of new areas. PeerJ Comput. Sci. **3**, e119 (2017)
23. Ruch, P., et al.: Using argumentation to extract key sentences from biomedical abstracts. Int. J. Med. Inform. **76**, 195–200 (2007)
24. Di Iorio, A., Nuzzolese, A.G., Peroni, S.: Towards the automatic identification of the nature of citations. In: CEUR Workshop Proceedings, pp. 63–74 (2013)
25. Athar, A., Teufel, S.: Context-enhanced citation sentiment detection. In: NAACL HLT 2012, pp. 597–601 (2012)
26. Do, H.H.N., Chandrasekaran, M.K., Cho, P.S., Kan, M.-Y.: Extracting and matching authors and affiliations in scholarly documents. In: ACM/IEEE-CS - JCDL 2013, pp. 219–228 (2013)
27. Lindsay, A., Read, J., Ferreira, J.F., Hayton, T., Porteous, J., Gregory, P.: Framer: planning models from natural language action descriptions. In: ICAPS, pp. 434–442 (2017)
28. Feng, W., Zhuo, H.H., Kambhampati, S.: Extracting Action Sequences from Texts Based on Deep Reinforcement Learning. arXiv:1803.02632 (2018)
29. Mei, H., Bansal, M., Walter, M.R.: Listen, Attend, and Walk: Neural Mapping of Navigational Instructions to Action Sequences. arXiv:1506.04089 (2015)
30. Pertsas, V., Christodoulou, T., Dallas, C., Constantopoulos, P., Papachristopoulos, L., Hughes, L.: Contextualized integration of digital humanities research: using the NeMO ontology of digital humanities methods. In: Digital Humanities 2016: Conference Abstracts, pp. 161–163. Jagiellonian University & Pedagogical University (2016)
31. Yeh, A.: More accurate tests for the statistical significance of result differences. In: COLING. vol. 2, pp. 947–953 (2000)